NEW SIDE-CHANNEL AND TECHNIQUES FOR HARDWARE TROJAN DETECTION

A PH.D. Dissertation
Presented to
The Academic Faculty

By

Luong N. Nguyen

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Georgia Institute of Technology

Georgia Institute of Technology

May 2020

NEW SIDE-CHANNEL AND TECHNIQUES FOR HARDWARE TROJAN DETECTION

Approved by:

Dr. Alenka Zajic, Co-advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Milos Prvulovic, Co-advisor
College of Computing
Georgia Institute of Technology

Dr. Raheem A. Beyah
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Angelos D. Keromytis
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Brendan D. Saltaformaggio
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Hyesoon Kim
College of Computing
Georgia Institute of Technology

Date Approved: March 26, 2019

”In the confrontation between the stream and the rock, the stream always wins, not through strength but by perseverance”

H. Jackson Brown

I dedicate this thesis to my loved ones. Thank you so much for being there for me.

Without your love and support, I would have quit long ago.

ACKNOWLEDGEMENTS

I would like to thank my advisors, Dr. Alenka Zajic and Dr. Milos Prvulovic, for the opportunity to work on this interesting topic. Their time, ideas, feedback, and support made the completion of this thesis possible. I also would like to thank my thesis committee: Dr. Raheem A. Beyah, Dr. Angelos D. Keromytis, Dr. Brendan D. Saltaformaggio, and Dr. Hyesoon Kim. Their time and inputs were essential in improving this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The thesis introduces a new physical side-channel, which we call the backscattering side-channel, and propose novel hardware Trojan (HT) and counterfeit integrated circuit (IC) detection techniques that exploit the backscattering side-channel. These techniques are capable of detecting different types of inactive HTs and counterfeit ICs on multiple circuit benchmarks while tolerating manufacturing variation.

For the last decade, demand for effective HT and counterfeit IC detection techniques has risen considerably. Numerous HT and counterfeit IC detection techniques have been published and side-channel analysis based approaches are among the most widely used. However, the problem with existing side-channels is that they do not provide enough resolution bandwidth, and information about the operation of electronic circuitry to detect small dormant hardware Trojan and small-changed counterfeit ICs. In addition, most previously proposed techniques do not take into account manufacturing variation, test on very few benchmarks, or rely on an unrealistic assumption of having a golden (HT-free or trusted-IC) sample. Motivated by these problems, our research focuses on introducing a new side-channel, i.e., the backscattering side-channel, and proposing novel techniques for HT and counterfiet IC detection using the new side-channel. We observe that the backscattering side-channel is especially suitable for HT detection because it has high bandwidth and spatial resolution, and its signal carries information about the current state of on-chip impedances.

To summarize, this work has 1) introduced a new backscattering side-channel, theoretically and experimentally proved the concept and existence of the side-channel, 2) developed new techniques for detection of dormant hardware Trojans and counterfeit ICs using the new backscattering side-channel, 3) modeled and compared the backscattering, electromagnetic (EM), and power side-channels and their performance in detecting software malware and hardware Trojans , 4) developed novel clustering based techniques that can

assist reverse engineering based methods for HT detection in a large population of integrated circuits, and 5) developed novel golden-chip-free HT detection techniques using backscattering side-channel using circuit impedance models.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Integrated circuits (IC) have become an integral aspect of our lives, by controlling most of electronic devices ranging from cellphones and washing machines to airplanes and rockets. Thus, the problem of ensuring authenticity and trust for ICs is critically important, especially for sensitive fields such as military, finance, and governmental infrastructure, and is gaining in importance as an increasing number of "things" become "smart" and connected into the Internet-of-Things (IoT). However, cost and time-to-market considerations have led IC vendors to outsource some, and in most cases many, steps in the IC supply chain. The sheer number and diversity of entities involved in modern IC supply chain, each with its own set of potentially malicious actors that can insert malicious modifications, referred as hardware Trojan (HT), in the IC [1], makes it difficult to trust the resulting ICs. In addition, it also leads to another severe security vulnerability, which is counterfeiting. The potential existence of HTs and counterfeit ICs significantly undermines the trust in any system that uses that IC, because the hardware usually provides the base layer of security and trust that all software layers depend and build on [2, 3, 4]. As a result, demand for effective HT and counterfeit IC detection techniques has risen considerably for the past few years.

For the last decade, numerous HT and counterfeit IC detection techniques have been published and side-channel analysis based approaches are among the most widely used. However, the problem with existing side-channels is that they do not provide enough resolution bandwidth, and information about the operation of electronic circuitry to detect small dormant hardware Trojan or counterfeit ICs with identical functionality with the authentic ones. In addition, most previously proposed techniques do not take into account manufac-

turing variation, test on very few benchmarks, or rely on unrealistic assumptions to detect HTs and/or counterfeit ICs.

Motivated by these problems, this research introduces a new physical side-channel, i.e., the backscattering side-channel, and proposes new HT and counterfeit IC detection techniques using this side-channel.

1.2 Creating a Backscattering Side-Channel to Enable Detection of Dormant Hardware Trojans

An hardware Trojan (HT) is a malicious modification of the circuitry of an integrated circuit. An hardware Trojan is completely characterized by its physical representation and its behavior. Typically, an HT is designed to be stealthy, so it only changes the functionality of the original circuit when specific conditions have been met. Thus the design of an HT typically has two key components: the *payload*, which implements the modification of the original circuit's behavior[1], and the *trigger*, which detects when the conditions for activating the payload have been met. The conditions that activate an HT occur very rarely, and until activated the payload is usually highly inert - it simply allows the IC to follow its original input/output behavior. This makes HTs extremely challenging to detect by traditional functional verification and testing - test inputs are unlikely to activate the HT, and without activation the HT has no effect on functional behavior of the IC. As a result, a plethora of counter-HT approaches have been proposed and they can be generally categorized into protection and detection techniques.

Protection techniques focus on making the IC resilient to the presence of HTs, i.e., on preventing the HT's payload from modifying the behavior of the IC, mostly by using fault-tolerance-inspired approaches to operate correctly even when an HT has been able to modify some of the internal signals. However, these techniques protect only certain parts of

[1]The HT's payload can also implement a non-functional change in the IC's behavior, e.g. to increase its power consumption, increase the IC's side-channel leakage of information, decrease its expected lifetime, etc.

the system, such as a bus [5] or on-chip interconnect [6], require redundant activity during normal operation [7], and/or rely on reconfigurable logic [8].

Most counter-HT techniques focus on detecting the presence of HTs. Some HT detection approaches are *destructive*, e.g., relying on successive removal of the IC's layers to scan the actual layout of the IC, reverse-engineer its GDSII and/or netlist-level design [9], and compare it to a trusted design. However, all the ICs that are found to be HT-free through such analysis are also destroyed by the scan, and the reverse-engineering is extremely expensive and time-consuming, so such destructive techniques can only be applied to a small sample of the larger population of IC.

Non-destructive HT detection approaches can be categorized according to whether they are applied to the design of the yet-to-be-fabricated IC (pre-silicon approaches), or to fabricated IC (post-silicon approaches). Pre-silicon approaches use functional validation, and code and gate-level netlist analysis [10, 11], but they cannot detect HTs that are inserted after the design stage, e.g., by editing the physical layout of the IC at the foundry. To overcome such concerns, post-silicon methods attempt to identify HTs in ICs received from the foundry.

Post-silicon non-destructive approaches detect HTs either through testing the functional properties of the IC, or by measuring non-functional (side-channel) behavior of the IC as it operates. Functional testing involves finding inputs that are likely to trigger unknown HTs that may exist in the IC, causing the payload of the HT to propagate the effects of the payload to the outputs of the IC, where they can be found to differ from expected outputs [12]. However, trigger conditions for HTs are designed to be difficult to reach accidentally, so the probability of detecting HTs is extremely low for conventional functional testing techniques. Additionally, functional testing techniques are likely to fail in detecting HTs whose payload does not change the input/output behavior or the IC, but rather causes increased power consumption, side-channel leakage of sensitive information, etc.

Among post-silicon approaches, HT detection through side-channel analysis appears

to be the most effective and widely used approach [13, 14]. These methods measure one or more non-functional properties of the IC as it operates, and compare these measurements to reference signals obtained through either simulation or measurement on a device known to be genuine. side-channels used by HT detection techniques include power consumption [15, 16, 17, 18], leakage current [19], temperature [20, 21], and electromagnetic emanations (EM) [22, 23, 24], and some approaches even combine measurements from multiple side-channels [25, 26].

Among side-channel-based HT detection approaches, some add the side-channel measurement capability to the chip, while others rely on measurements that are external to the chip itself. With on-chip measurements, the measurement circuitry is added to the design [27, 28, 29], which allows the specific chosen signals to be measured close to the signal's source. However, the additional circuitry for measurements, and for routing the desired signals to the measurement circuitry, impacts chip size, manufacturing cost, performance, and power, and this impact increases as the set of individually measurable signals increases.

Finally, external-measurement side-channel techniques require no modifications to the IC itself, and instead rely on externally observable side-effects of the IC's normal activity. Since an HT is typically much smaller than the original circuit, an ideal side-channel signal would have little noise and interference so that the HT's small contribution to the signal is not obscured by the noise. Additionally, the HT's payload is largely inert until activated, and activation during measurement is highly unlikely, so ideally the side-channel signal would be affected by the presence of the payload circuitry, even when it is inert. Finally, before activation, what little switching activity the HT does create is in its trigger component, which usually has only brief bursts of switching when the inputs it is monitoring change. Thus an ideal side-channel signal would have high bandwidth, such that these brief bursts of current fluctuation due to switching activity in the HT can be identified.

Unfortunately, as mentioned earlier, existing externally-measurable side-channel sig-

nals, such as temperature, voltage, power consumption, and electromagnetic emanations [22], tend to vary mostly in response to current variation due to switching activity. However, temperature changes slowly and has very limited bandwidth, and voltage and supply current have low bandwidth [24] because on-chip capacitances that help limit supply voltage fluctuation act as a low-pass filter with respect to both current and voltage as seen from outside the chip. Electromagnetic emanations can have high bandwidth, but their signal-to-noise ratio is affected by noise and interference.

Motivated by the above-mentioned drawbacks of previous techniques, this thesis introduces a new physical side-channel, i.e., the backscattering side-channel, that is created by transmitting a signal toward the IC, where the internal impedance changes caused by on-chip switching activity modulate the signal that is backscattered (reflected) from the IC. To demonstrate how this new side-channel can be used to detect small changes in circuit impedances, we use it to implement a new proof-of-concept method for non-destructively detecting HTs from outside of the chip. To our knowledge, this is the first off-chip side-channel technique capable of detecting *inactive* HTs while tolerating variations that exist across hardware instances. Also, to our knowledge, backscattering has never before been used as a side-channel signal to infer information about the operation of electronic circuitry, even though backscattering has been used extensively for RFID tags and other short-range communications [30].

We observe that backscattering not only can be used as a side-channel signal, but also that it is especially suitable for HT detection because the backscattered signal carries information about the current state of on-chip impedances, unlike traditional side-channels that carry information about brief changes in current. Furthermore, like the traditional EM side-channel, the backscattering side-channel has high bandwidth but, unlike the traditional EM signal, the strength of the backscattered signal can be increased when needed, its frequency can be shifted to avoid noise, interference, and poor signal propagation conditions, and it can be more accurately focused on a specific part of the chip.

We test our new HT detection technique using multiple HTs from the Trusthub benchmark [31] and show that it is highly accurate in detecting even *inactive* HTs while avoiding false positives. We compare our approach to one that applies the same signal analysis to traditional electromagnetic emanations, and our results confirm backscattering yields a dramatic improvement in HT detection accuracy. We further evaluate the sensitivity of our approach by separately reducing the size of the HT's trigger and payload components, and showing that HT detection of inactive HTs largely depends on the size of the trigger component, and that our approach can detect even HTs with significantly reduced triggers. Additionally, we also evaluate how our approach is affected by manufacturing and other variations, by using different physical instances of the same design for training and testing, and find that the technique largely maintains its ability to detect HTs accurately even when trained on only one instance and used to test another.

1.3 A Comparison of Backscattering, EM, and Power Side-Channels and Their Performance in Detecting Software and Hardware Intrusions

Side-channel analysis is a powerful tool from both an attacker's and defender's perspective. Attackers use side-channels to circumvent traditional access controls and protections by exploiting the observable side effects of computation rather than attacking the computation's functionality. Computations have many observable side effects through an analog medium, such as power consumption [32, 33, 34], sound [35, 36], and electromagnetic emanations [37, 38, 39], that can be exploited to create side-channel attacks. A number of studies have been published on preventing side-channel attacks and the leakage of sensitive information. For example, several countermeasures for protecting video displays and smart-cards from EM leakage involve low-cost shielding techniques, the use of asynchronous circuits, or changing the layout of circuitry [40, 41, 42, 43, 44]. Additionally, research has demonstrated methods for systematically identifying and quantifying EM side-channel signals [45, 46, 47, 48].

Defenders use side-channels for tracking program activities on various code levels such as loops, paths, basic blocks, and individual instructions [49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59], as well as for hardware Trojan detection [13]. Side-channel analysis for tracking program activities relies on monitoring a device's power or EM fluctuations and relating them to software activities on a device's processor. This analysis has been used for protecting against attacks targeting the battery life of hand-held mobile devices [49], for integrity assessment of Software Defined Radios [50], for malware detection on embedded medical devices [60], IoT (Internet of Things) devices [61, 62, 63], etc. Side-channel analysis for HT detection relies on measuring non-functional properties from outside the integrated circuit (IC), and comparing the measurements to reference signals produced by either simulation or a golden example. Side-channels used for HT detection include power consumption [15, 16, 18, 17], leakage current [19], temperature [20, 21], EM [22, 23, 24], a combination of multiple side-channels [25, 26], and more recently backscattering side-channels [64]. The backscattering side-channel is a consequence of impedance changes in switching circuits. One example of such a side-channel is when digital logic activity causes incoming EM signals to be modulated as they are reflected (backscattered) at frequencies that depend on both the incoming EM signal and the circuit activity.

All previous work indicates that backscattering, EM, and power side-channel analysis can be a powerful tool for both attackers and defenders; however, it is not clear which type of side-channels provides the most useful information. Hence, the objective of this work is to model and quantitatively compare the backscattering, EM, and power side-channels and their performance in detecting malware and HTs. We start by describing the backscattering side-channel and comparing it with EM and power side-channels, two of the more widely used types of side-channels. Then, we characterize, model, and compare spectral characteristics of all three side-channels. Finally, we compare the performance of all three side-channels in detecting malware and HTs. The results show that for larger changes in the signals, such as those caused by malware intrusions, all three side-channels perform

similarly. However, when smaller changes need to be observed, such as those caused by HTs, the backscattering side-channel outperforms the EM and power side-channels.

1.4 A Novel Golden-Chip-Free Clustering Technique Using Backscattering Side-Channel for Hardware Trojan Detection

As we discussed earlier, over the past few years, a significant shift in the manufacturing model and design flow of IC companies has been observed due to various factors including time-to-market, cost reduction demands, and the increased complexity of ICs. These companies have fully adopted the "horizontal model", in which they use IPs from third-party companies and outsource all hardware fabrication to offshore foundries. While the new design flow model allows for reduction in the cost, time-to-market and fabrication errors, it raises questions on the hardware level trust because HTs could be injected into an IC by adversaries at any stage of the design and fabrication flow. HT insertion at the foundry is the most common scenario because IC companies fabricate their chips in offshore foundries, which are harder to secure. Hence, numerous HT detection techniques are proposed to detect HT insertion at the foundry stage. These techniques can be classified into two groups: reverse engineering and side-channel approaches.

Reverse-engineering techniques rely on destructive scanning the actual IC layout to re-build the GDSII and netlist level of the chip [9]-[70]. The destructive scanning process consists of decapsulation to remove the die from the package, de-layering to strip each layer off the die, and imaging to reconstruct images for every layer. After getting the GDSII and netlist level of the chip, these techniques are capable of detecting any malicious post-RTL-design insertion with very high accuracy by comparing them to the GDSII and netlist of a trusted design. However, reverse-engineering is extremely time-consuming, expensive and destructive because of chip demolishing after reverse engineering. Therefore, applying reverse engineering based HT detection techniques to test a large population of ICs, although accurate and reliable, it is not practical.

On the other hand, side-channel analysis based approaches rely on measuring some non-functional properties from outside of the IC while it operates, and comparing the measurements to reference signals produced by either simulation [22]-[72] or by a "golden-sample" device [73]. Potential side-channels include backscattering [73], power consumption [15], [16], leakage current [74], temperature [20], electromagnetic emanations (EM) [22], [24], or a combination of multiple side-channels [25], [26]. In some techniques, additional measurement circuitry is added to the design [28], [29], which allows the specific signals to be measured close to the signal source. However, additional circuitry results in circuit size, manufacturing cost, performance, and power overhead. Therefore, the majority of side-channel based detection techniques require no modifications to the chip itself, and rely on measuring side-channel signals outside of the chip. In contrast to reverse engineering techniques, the side-channel based techniques can be applied to a large population of ICs because side-channel measurements do not require damaging the board while conducting testing. However, the disadvantage of side-channel techniques is their dependence on either having a "golden" (HT-free) chip, which is not a practical assumption for foundry-inserted HTs in single-source ICs, or having a detailed simulation model, which is often impractical (complex ICs, 3rd-party IP, etc.).

To overcome these shortcomings of both types of approaches, we propose a novel "golden-chip-free" clustering algorithm using backscattering side-channel. This technique is bridging the gap between destructive reverse-engineering and traditional side-channel detection techniques. The proposed clustering algorithm clusters a large population of ICs based on the effect of a hypothetical HT would have on the backscattering side-channel signal. In practical terms, the technique creates clusters such that the ICs in each cluster can be considered equivalent in terms of presence or absence of an HT. This allows reverse-engineering of one IC in each cluster to be used to assess the status (in terms of HT presence and nature) of that entire cluster.

A number of techniques utilizing clustering algorithms for HT detection have been

previously proposed [75]-[77], however, the majority of these methods are pre-silicon approaches, which means that they can not detect HTs inserted in the fabrication stage [75]-[76]. A post-silicon clustering technique using side-channel analysis has been proposed in [77], but authors only test their method on a set of two FPGAs, which does not give enough statistics to evaluate manufacturing variations among different hardware instances. In addition, the technique uses power side-channel, which provides very limited resolution and bandwidth [73]. Unlike these previous approaches, our technique works for HTs inserted at foundries without needing a golden chip or any a priori knowledge of the chip circuitry. We have tested the proposed technique on a set of 100 boards which provides enough statistics for manufacturing variation and show that our technique outperforms other side-channels for HT detections [73]. We evaluate our clustering algorithm for multiple HT and circuit benchmark designs over a set of 100 boards, in which each board will be randomly loaded with either a HT-free or an HT-infected design. In all these experiments the HT (if present) is in a dormant state, i.e., none of the HTs are activated during this evaluation. The results show that our technique is capable of clustering all boards correctly for 9 different Trojan designs on 3 different benchmark circuits from Trusthub [31] with 100 $\%$ accuracy. In additional experiments, we make HTs more stealthy by reducing the size of their trigger, resulting in trigger circuits that are as small as 0.19% of the original circuit, and find out that our technique still correctly clusters the boards. The following summarizes the contributions of this work:

- This work describes a novel clustering algorithm that is capable of classifying a large population of ICs into clusters without having a "golden" (known-to-be-HT-free) chip, and with no a priori knowledge about circuitry of the chip. The algorithm is based on clustering spectral features of backscattering side-channel that tend to be the most impacted by dormant HTs.

- This work describes a testing environment that includes a set of 100 boards and implemented multiple HT benchmarks. This large set of boards allows a thorough

evaluation of the manufacturing variation among different hardware instances with enough statistics, which has not been done before.

1.5 Counterfeit IC Detection Using Backscattering Side-Channel

A counterfeit IC is an illicit copy of a legitimate chip, typically with some difference in terms of performance, characteristics, or materials, but which is sold or used as a legitimate (authorized) IC [78]. Counterfeiting of ICs has become a major challenge for the semiconductor industry, in large part because existing test techniques and protection mechanisms are not very effective in detecting counterfeit ICs. Unfortunately, over the past few decades the problem has been getting worse, because globalization of the semiconductor supply chain has led companies to outsource many steps of their integrated circuit (IC) production cycle, and incidences of counterfeit ICs have increased rapidly. In 2015, it was reported that the illicit production of counterfeit ICs has cost IC companies $100 billion [79, 80], and that this cost has steadily increased. This has been, and still is, a significant threat to the IC industry, not only because it negatively impacts innovation and economic growth, but also because it represents a serious threat/risk for systems that incorporate these counterfeit ICs. In practice, counterfeit ICs have found their way into almost all industrial sectors, including ones that are highly sensitive to potential security, reliability, and other risks: cloud infrastructure, finance, government infrastructure, military systems, etc. As a result, the need for effective detection of counterfeit ICs has increased tremendously.

Over the past few years, a plethora of papers have been published on the topic of counterfeit ICs. Such work can be roughly divided into *detection* and *avoidance* techniques [81]. Avoidance aims to make counterfeits easily detectable, e.g. by adding circuitry to legitimate ICs to act as a signature/watermark [78, 82, 83], by fabricating different parts of the chip layout in different foundries [84], etc. However, avoidance techniques significantly add to the cost of an IC, which prevents them from widespread adoption.

In contrast, techniques for detection of counterfeit IC detection focus on distinguish-

ing counterfeit ICs from authentic ones, usualy without adding circuitry to the IC, changing its layout, etc. Detection techniques can be based on either physical tests or electrical tests [85]. Physical tests rely on examining the physical and chemical/material properties of the IC's package, leads, and die in order to detect procedural, mechanical, and environmental deviations in counterfeit ICs [84]. These techniques include external visual inspection (EVI), X-ray imaging, resurfacing, microscopy scanning, material analysis such as X-Ray Fluorescence (XRF), Fourier transform infrared spec. (FTIR), ion chromatography [86, 87], etc. While physical tests can, in principle, be used to detect all types of counterfeit ICs, the more reliable such tests are destructive, time-consuming, and expensive [85, 84].

Electrical tests consist of parameter tests, function tests, curve tracing, built-in tests and structural tests [79, 88, 89]. Unlike physical tests, electrical tests are non-destructive, relatively fast, and inexpensive. However, electrical tests rely on determining whether the IC's functionality is correct, which does not detect counterfeit ICs that have the same functionality but different layout as authentic ones. In addition, reliable detection of counterfeits typically necessitates use of a number of electrical test techniques, some of which require extra circuitry to be added to the design, so the total added IC cost for all these techniques can be significant.

Motivated by the above-mentioned drawbacks of previous detection techniques, this work proposes a novel non-destructive and fast technique using the backscattering side-channel for detection of counterfeit ICs with the same functionality, but different layout, with authentic ones. This includes ICs that contain hardware Trojan horses, such that the functionality of the counterfeit IC is identical to that of a legitimate IC, except under very specific conditions that are highly unlikely to be encountered during electrial testing. We choose to use backscattering side-channel, a new physical side-channel that has been demonstrated to outperform other side-channels in terms of detecting hardware Trojan horses (HTs) in ICs [64]. However, in this work we focus on detecting changes in IC layout and placement, i.e. detecting counterfeit ICs that are functionally exact equivalents

of legitimate ICs, without any additional logic gates (or connections between those gates).

1.6 Golden-Chip-Free Hardware Trojan Detection Technique Using Backscattering Side-Channel

As we discussed earlier, side-channel analysis approaches are the most widely used HT detection techniques. They have advantages of being non-destructive and relatively fast, which is suitable for testing a large number of ICs. However, the main drawback of most existing side-channel HT detection techniques is the dependence on having a golden (HT-free) chip for training [20, 23, 18]. The assumption of having a golden sample is too strong, and often unrealistic, which prevents them from being used for practical deployments of HT detection.

There are few papers that propose to circumvent this problem by using simulation and/or modeling in lieu of a golden chip. In [22], the authors present a method using EM side-channel to detect HTs without having to have a golden circuit and the technique was tested with multiple different Trojans from Trusthub. However, in the paper, the authors use the EM side-channel, which is proved to have multiple shortcomings when used for HT detection [64]. Furthermore, hardware Trojans were activated in the experiments. This is not practical because it is extremely difficult to activate Trojans without a priori knowledge of their triggering mechanism. As a result, it is not clear that how their technique would perform with dormant hardware Trojans and its applicability to larger and more complicated circuits.

Another post-silicon technique utilizing side-channel analysis without the need of having a golden-free-chip has been proposed in [77], but the technique relies on using power side-channel, which provides very limited resolution and bandwidth as discussed above. In addition, [75] presents an information-theoretic approach that estimates the statistical correlation between the signals in a design and then use a weight normalization and clustering algorithm to detect HTs. In [11], the authors propose COTD, a HT detection tech-

nique based on analyses of the controllability and observability of gate-level netlist and utilizing an unsupervised clustering to detect HTs by exploiting significant inter-cluster distance caused by the controllability and observability characteristics of the Trojan gates. In [76], a technique based on "outliers", is proposed to identify suspicious signals in a netlist, and clustering technique to detect HTs. However, all of these methods are pre-silicon approaches, which means that they can not detect HTs inserted in the fabrication stage.

Motivated by the shortcomings of the previous techniques, this work proposes a novel golden-chip-free hardware Trojan detection technique using backscattering side-channel with circuit impedance models. As we mentioned earlier, backscattering side-channel outperforms other side-channels in hardware Trojan detection because it is impedance-based, which means it gives information about the impedance change inside the chip. When a HT is attached to the circuit, it changes the circuit impedance, regardless of whether it is activated or not. We build models that help calculate the reference impedances of the benchmark circuits and estimate the expected power of the backscattered signal of clock harmonics. Then we compare them against the measurements to detect HTs without having to have an golden sample.

We start with simple circuits such as a transistor, then build up to impedance models for more complicated circuits. These models are used to calculate the reference impedances of the circuits, and estimate reference power of clock harmonics. Then we use them for HT detection. Our algorithm reports a design as Trojan-free if the real measurements matches its model's reference, and report a design as Trojan-infected otherwise. We test our technique on multiple Trojan benchmarks and the results show that our technique can detect Trojan with 100% accuracy and 0% false positives, if the Trojan trigger is big enough.

1.7 Research Contributions

The research contributions of this thesis are

- Introduce a new physical side-channel, i.e., the backscattering side-channel, that is created by transmitting a signal toward the IC, where the internal impedance changes caused by on-chip switching activity modulate the signal that is backscattered (reflected) from the IC [64].

- Theoretically and experimentally prove the existence and the concept of the new backscattering side-channel. Demonstrate measurements for the backscattering side-channel [64].

- Propose a new method for non-destructively detecting HTs from outside of the chip using the backscattering side-channel [64]. To our knowledge, this is the first off-chip side-channel technique capable of detecting *inactive* HTs while tolerating variations that exist across hardware instances. Also, to our knowledge, backscattering has never before been used as a side-channel signal to infer information about the operation of electronic circuitry.

- Model and quantitatively compare backscattering, electromagnetic (EM), and power side-channels and discuss the performance of these three side-channels for detecting software malware and hardware Trojans [90].

- Describe a novel clustering algorithm that is capable of classifying a large population of ICs into clusters without having a "golden" (known-to-be-HT-free) chip, and with no a priori knowledge about circuitry of the chip. The algorithm is based on clustering spectral features of backscattering side-channel that tend to be the most impacted by dormant HTs [91].

- Describe a testing environment that includes a set of 100 boards and implemented multiple HT benchmarks. This large set of boards allows a thorough evaluation of the manufacturing variation among different hardware instances with enough statistics, which has not been done before [91].

- Propose a novel method that uses the backscattering side-channel to cluster ICs such that counterfeits are separated from legitimate ICs [92].
- Build circuit impedance models and demonstrate a golden-chip-free hardware Trojan technique using backscattering side-channel by relying on circuit impedance models.

1.8 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 presents background of HTs, counterfeit ICs, and the side-channels. Chapter 3 discusses creating a backscattering side-channel to enable detection of dormant hardware Trojans. Chapter 4 presents modeling and comparison of backscattering, EM, and power side-channels and their performance in detecting software and hardware intrusions. Chapter 5 proposes a novel golden-chip-free clustering technique using backscattering side-channel for hardware Trojan detection. Chapter 6 presents counterfeit IC detection using backscattering side-channel. Chapter 7 proposes a golden-chip-free hardware Trojan technique using backscattering side-channel by relying on circuit impedance models. Finally, chapter 8 summarizes thesis contributions, presents possible future directions for related research, and concludes the thesis.

CHAPTER 2

BACKGROUND

2.1 Hardware Trojans

2.1.1 The Emerging Thread of Hardware Trojan

Most software systems are built on the assumption that the underlying hardware can be trusted to perform the requested operations correctly, and even when incorrect hardware behavior is considered, it is assumed to be erroneous rather than malicious. HTs break this assumption, so the potential presence of unknown HTs in the system's hardware effectively eliminates trust in the overall system regardless of how trustworthy the system's software is. Over the past several years, numerous papers have been published on the topic of understanding the intent, behavior [14, 93], and implementation of HTs [94, 95, 96, 31]. Several studies have focused on characterizing and classifying HTs [97, 13, 98, 31] according to activation mechanism, functionality, location on the IC, the point in the IC design cycle and supply chain at which they are inserted, etc.

A common characteristic of HTs is that they are designed to avoid detection, so they activate their malicious activity rarely to avoid being relatively easily detected, e.g., during functional testing of the IC [93]. Therefore, a typical HT consists of a *trigger* circuit and *payload* circuit, as illustrated in Fig. 2.1. The trigger circuit is monitoring a set of signals to detect when the conditions for activation of the payload have been met, while the payload implements the actual malicious functionality. The malicious activity can be functional, e.g., when the HT's output modifies the outputs of the overall circuit to cause harm or leak sensitive information, and/or non-functional, e.g., when the payload increases power consumption, causes excessive wear-out to reduce the lifetime of the IC, leaks sensitive information through a side-channel, etc.

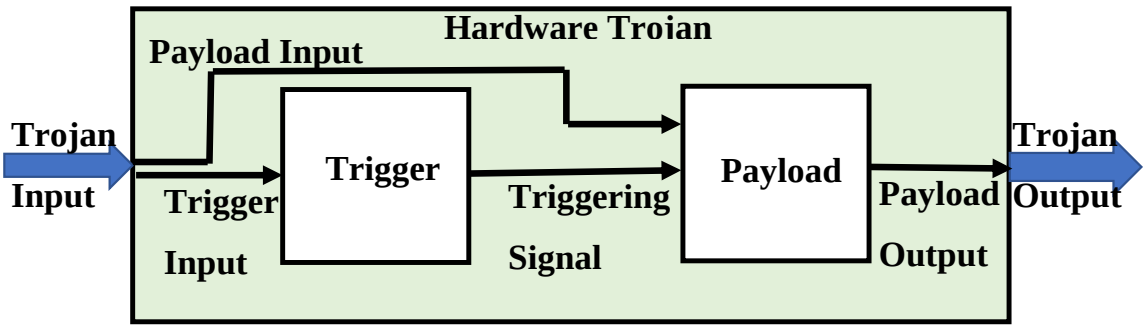


Figure 2.1: Simplified Block Diagram of an HT.

2.1.2 Adversaries and Attacks

The life cycle of an IC is depicted on the left side of Fig. 2.2. Ideally, all of the steps in this life-cycle would be performed by a single trusted entity, which would design, fabricate, test, package, and deploy the IC. However, cost-reduction, time-to-market, IC complexity, and other considerations have recently led companies to specialize in a single step in the IC design and/or manufacturing, so the overall IC is typically designed by one entity, usually includes intellectual property (IP) blocks of several other entities and design tools from yet another entity, is fabricated, tested, and packaged by one or more other entities, and is finally deployed by yet another entity. Different parts of the life cycle typically also take place in several different countries. Technically, HTs could be injected to an IC by adversaries at any stage of its design and fabrication flow.

A subset of opportunities for inserting HTs into the IC is shown on the right side of Fig. 2.2, but technically, HTs could be injected to an IC by adversaries at any stage of its design and fabrication flow.

2.1.3 Hardware Trojans: Taxonomy

As the number and complexity of HTs increased dramatically, several studies on the topic of characterizing and classifying HTs have been published over the last few years [31]-[97]. The most comprehensive work to date is proposed by [31]. Figure 2.3 illustrates different

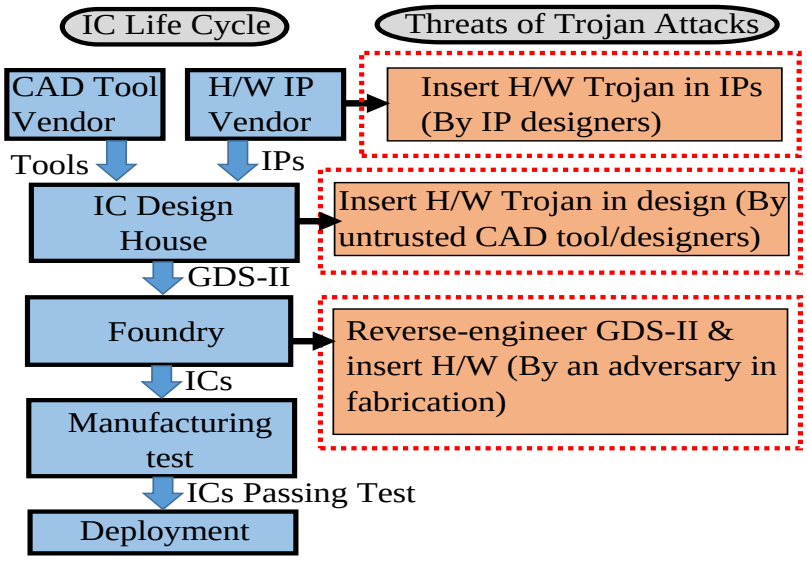


Figure 2.2: Examples of HT insertion activity in IC life cycle [93].

ways of classifying HTs. As shown in the figure, HTs can be classified by their activation mechanism, functionality, or the phase in the IC design flow they are inserted into the chip.

2.2 Counterfeit IC

A counterfeit IC is an illicit copy of a legitimate chip, typically different in terms of performance, characteristics, and/or material, but which is sold and/or used as if it was a legitimate (authorized) IC [78]. There are three major categories of counterfeit ICs: remarked/recycled ICs, out-of-spec/defective ICs, and cloned ICs [78]. The first group includes aged ICs sold as new, ICs remarked with forged information to mimic more expensive (e.g. higher-rating) ICs, etc. The second group includes out of specification ICs, ICs that were rejected during manufacturing tests but sold as normal ones, and ICs that have been tampered with during manufacturing (e.g. to infect them with a hardware Trojan horse). The last group includes overproduced ICs, and unauthorized production of an IC by illegally obtaining the design of the IC either at RTL level, netlist level, or layout level.

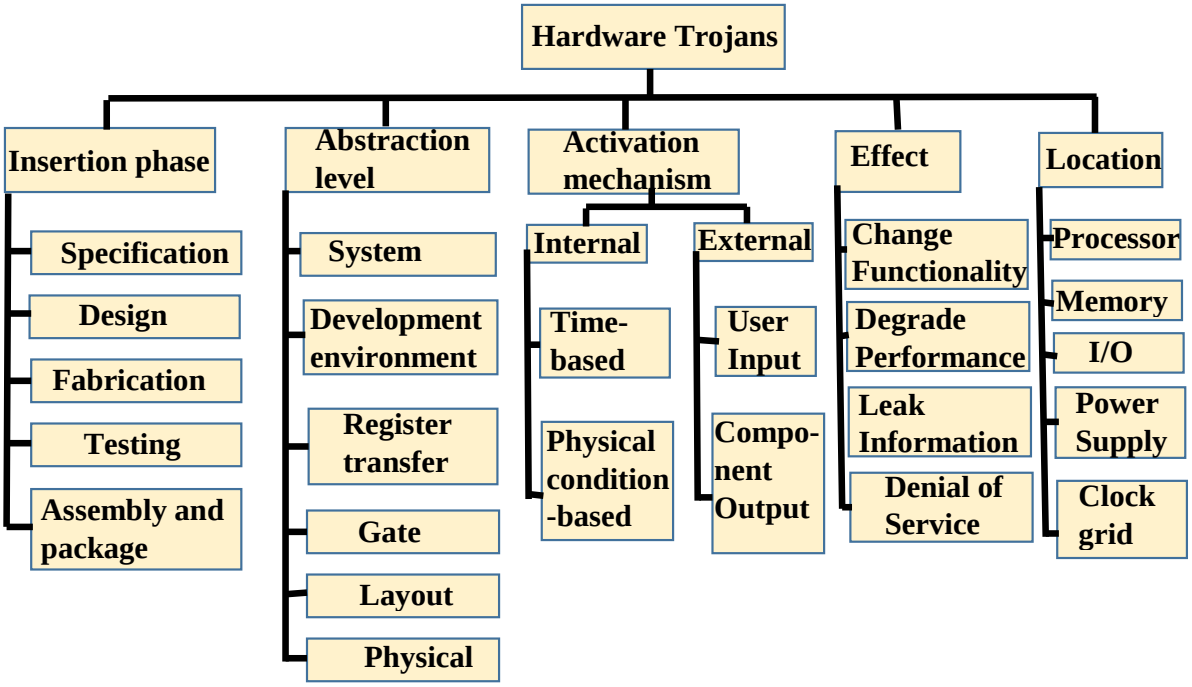


Figure 2.3: Hardware Trojans Taxonomy [31].

2.3 Backscattering

The backscattering concept has been used to enable RFID tags to transmit information with very low energy expenditure [30]. A typical RFID system based on backscattering is illustrated in Fig. 2.4. The data transmission requires the RFID reader to emit a continuous wave (an RF signal at some frequency f_c) toward the RFID tag. The RFID tag contains an antenna that can be connected to one of two impedances, Z_0 or Z_1, one of which is chosen to maximize the antenna's reflection coefficient (also called radar cross-section, or RCS) for frequency f_c, while the other impedance is chosen to minimize the antenna's RCS for f_c. The RFID tag typically contains an application-specific integrated circuit (ASIC) chip that can electronically switch the antenna's connection between these two impedances, which modulates the signal that reflects (backscatters) from the antenna according to the data bits the RFID tag wishes to transmit. The RFID reader then receives and demodulates the backscattered signal to retrieve the data transmitted by the tag. This enables use of very compact RFID tags, because the energy for the signal "transmitted" by the RFID tag is

entirely provided by the RFID reader [1].

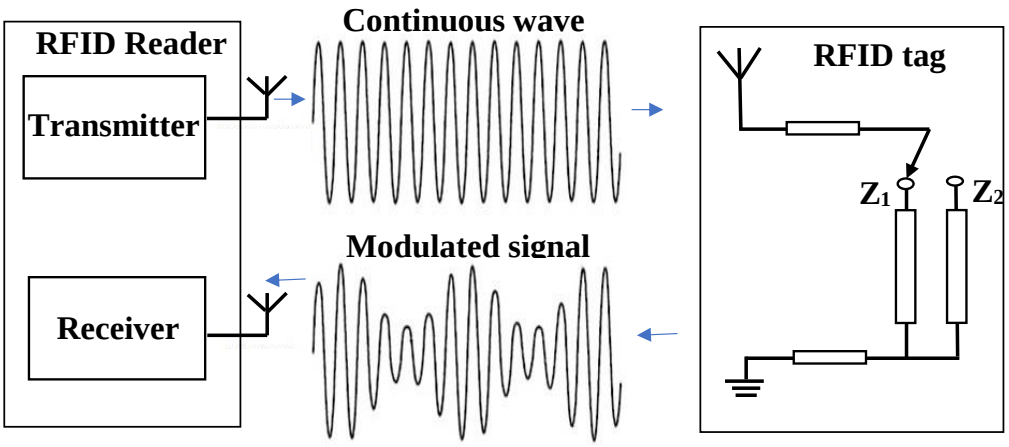


Figure 2.4: An illustration of backscatter data communication.

2.4 Traditional Analog Side-Channels

2.4.1 EM Side-Channels

In EM side-channels, the variation in the current-flow in a device while it is in operation causes the device to emit EM waves [99]. An advantage of EM side-channels is that they have a large bandwidth. Furthermore, they allow an attacker to monitor the device from a distance; however, the range is limited by the magnitude of the radiation. Since the strength of the radiation is a consequence of the physical properties of the device, the attacker has little control over it. As a result, the difficulty of monitoring the EM side-channel can vary greatly between different types of devices and programs.

2.4.2 Power Side-Channels

In power side-channels, information about what is being performed by a device can be obtained by monitoring how its power consumption varies. Because some power is consumed by a transistor while it is active, the device's power draw is directly related to its transistors'

[1] Typically the electronic switching done by the RFID tag's ASIC is powered by energy-harvesting using the reader's signal, which completely eliminates the need for long-term energy storage (e.g. a battery) in the RFID tag.

activity [100]. One weakness of power side-channels is that they require a direct connection to the monitored device. Furthermore, power side-channels have a limited bandwidth [24] because the on-chip mechanisms that limit the supply voltage fluctuations also act as a low-pass filters with respect to the current and voltage that can be measured at the outputs of the chip.

CHAPTER 3
CREATING A BACKSCATTERING SIDE-CHANNEL TO ENABLE DETECTION OF DORMANT HARDWARE TROJANS

3.1 Overview

As discussed in Chapter 1, HT detection through side-channel analysis is the most effective and widely used HT detection approach. Among side-channel-based HT detection approaches, external-measurement side-channel techniques are generally preferred because they require no modifications to the IC itself, which means there is no degrade on chip size, manufacturing cost, performance, and power. Since a HT is typically much smaller than the original circuit, an ideal side-channel signal would have little noise and interference so that the HT's small contribution to the signal is not obscured by the noise. Additionally, the HT's payload is largely inert until activated, and activation during measurement is highly unlikely, so ideally the side-channel signal would be affected by the presence of the payload circuitry, even when it is inert. Finally, before activation, what little switching activity the HT does create is in its trigger component, which usually has only brief bursts of switching when the inputs it is monitoring change. Thus an ideal side-channel signal would have high bandwidth, such that these brief bursts of current fluctuation due to switching activity in the HT can be identified.

Unfortunately, as mentioned earlier, existing externally-measurable side-channel signals, such as temperature, voltage, power supply current, and electromagnetic emanations [22], tend to vary mostly in response to current variation due to switching activity. However, temperature changes slowly and has very limited bandwidth, and voltage and supply current have low bandwidth [24] because on-chip capacitances that help limit supply voltage fluctuation act as a low-pass filter with respect to both current and voltage as

seen from outside the chip. Electromagnetic emanations can have high bandwidth, but their signal-to-noise ratio is affected by noise and interference.

Motivated by the above-mentioned drawbacks of previous techniques, this work introduces a new physical side-channel, i.e., the backscattering side-channel, that is created by transmitting a signal toward the IC, where the internal impedance changes caused by on-chip switching activity modulate the signal that is backscattered (reflected) from the IC. To demonstrate how this new side-channel can be used to detect small changes in circuit impedances, we use it to implement a new proof-of-concept method for non-destructively detecting HTs from outside of the chip.

We test our new HT detection technique using multiple HTs from the Trusthub benchmark [31] and show that it is highly accurate in detecting even *inactive* HTs while avoiding false positives. We compare our approach to one that applies the same signal analysis to traditional electromagnetic emanations, and our results confirm backscattering yields a dramatic improvement in HT detection accuracy. We further evaluate the sensitivity of our approach by separately reducing the size of the HT's trigger and payload components, and showing that HT detection of inactive HTs largely depends on the size of the trigger component, and that our approach can detect even HTs with significantly reduced triggers. Additionally, we also evaluate how our approach is affected by manufacturing and other variations, by using different physical instances of the same design for training and testing, and find that the technique largely maintains its ability to detect HTs accurately even when trained on only one instance and used to test another.

The rest of this chapter is organized as follows. Section 3.2 introduces the hypothesis and proof of existence of the backscattering side-channel. Section 3.3 summarizes the advantages of the new backscattering side-channel, compared with existing traditional side-channels. Section 3.4 presents a proof of how hardware Trojan can be detected by using the backscattering side-channel. Section 3.5 defines our detection technique and algorithm, while Section 3.6 describes the Trojans we use, how we implement those hardware Trojans

on a FPGA, and the measurement setup. Section 3.7 evaluates the effectiveness of our technique in cross-training scenarios, how the size and position of HT's trigger and payload affect detection accuracy, and the difference in HT detection when the Trojans are dormant versus when the Trojans are activated. Section 3.8 further evaluates the robustness of the technique, by testing it on multiple boards with multiple HT designs. Finally, Section 3.9 concludes this chapter.

3.2 Exploiting Backscattering as a New Physical Side-Channel

The motivation to explore backscattering as a side-channel was a hypothesis that the backscatter radio effect should be present in electronic devices. Specifically, transistors in digital circuits switch between two states (closed and open), which changes the impedances connected to wires within the IC, which should modulate a signal that is backscattered from the IC. An example of this is shown in Fig. 3.1 for a CMOS NOT gate, which consists of two pull-up transistors connected in parallel and two pull-down transistors connected in series, as shown in Fig. 3.1 (a).

Depending on its output (logical 1 or logical 0), the NOT gate exhibits two impedance states shown in Fig. 3.1, where R_1 is the resistance of the in-parallel connection of conducting (turned-on) pull-up transistors, while R_0 is the in-series connection of conducting (turned-on) pull-down transistors. Thus the impedances "seen" from the gate's V_{DD} and ground connections change depending on the output state of this gate, and unless the transistor geometry and doping levels are perfectly chosen to make R_1 and R_0 be exactly the same, the impedances "seen" from the gate's output will also change with the gate's output state [101]. Furthermore, actual impedances also have parasitic capacitances and inductances that depend on the exact geometry of the gate and its connections, making it highly likely that the overall impedances change with the gate's output state.

Other types of gates exhibit similar state-dependent impedance changes, so when a continuous-wave signal is transmitted toward a set of gates, the backscattered signal can

be expected to change as the gates' states change, thus creating an *impedance-based* side-channel, in contrast to the traditional EM side-channel which is current-flow based.

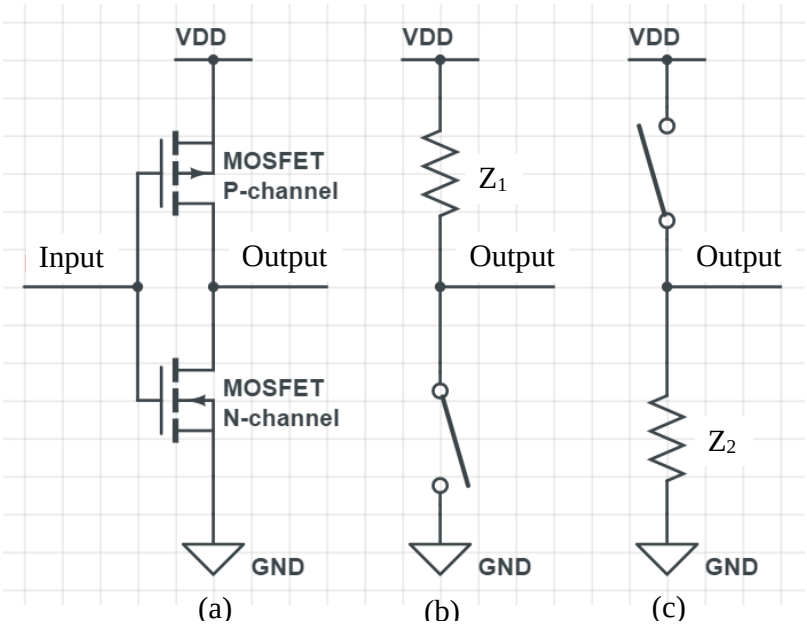


Figure 3.1: CMOS NOT gate (a) and its two equivalent impedance circuits (b).

To illustrate how this concept works in practice, The author implements a ring of flip-flops as shown in Fig. 3.2 in an Altera DE0 board with a Cyclone V FPGA. The flip-flops are initialized with alternating values, such that each flip-flip toggles from 0 and 1 and back again with a frequency of f_m. Fig. 3.3 shows the resulting output voltage of a flip-flop in this ring, which has a square-wave pattern with frequency f_m.

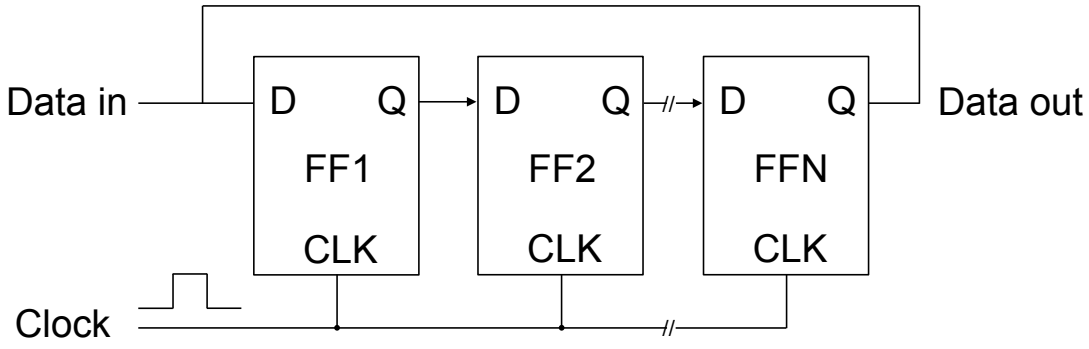


Figure 3.2: Cyclical shift register.

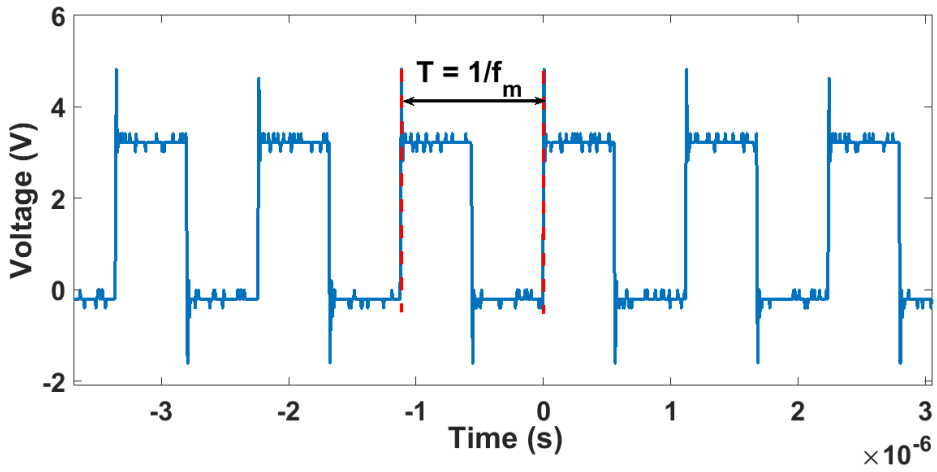


Figure 3.3: Measured voltage at the output of flip-flops switching at f_m=900 kHz.

A continuous wave (sinusoidal) signal at frequency $f_{carrier}$ is transmitted toward the FPGA chip, and receive the backscattered signal using the same setup as in Fig. 3.9.

The backscattered signal, if it is modulated by the switching activity, should contain not only a component at $f_{carrier}$, but also side-band components at frequencies $f_{carrier} - f_m$ and $f_{carrier} + f_m$. The $f_{carrier}$=3.031 GHz in this experiment was chosen to avoid interference from other periodic signals on the DE0-CV board, e.g. the crystal-oscillator-controlled 50 MHz clock and its harmonics. To ensure that the side-channel created by the backscattering effect corresponds to on-chip activity, none of the flip-flop outputs is used to control any off-chip activity, and all of the FPGA chip's output pins are kept in a constant state throughout the experiment.

Fig. 3.4 plots the spectra of the backscattered signal in this experiment. The first spectrum was collected for f_m=900 kHz. This spectrum contains a strong component at $f_{carrier}$, which represents the unmodulated part of the backscattered (reflected) signal, and also side-band signals 900 kHz to the left and to the right of $f_{carrier}$. These side-band signals are a consequence of the carrier signal being modulated by on-chip toggling activity through the backscattering effect. To further increase confidence that these side-band signals are indeed a consequence of the backscattered signal being modulated by on-chip toggling, we change the f_m to 1.2 MHz, and observe that the spectral component at $f_{carrier}$ remains at the same

frequency, the frequencies of side-band components change with f_m as predicted by the modulation hypothesis (sidebands at $f_{carrier} \pm f_m$).

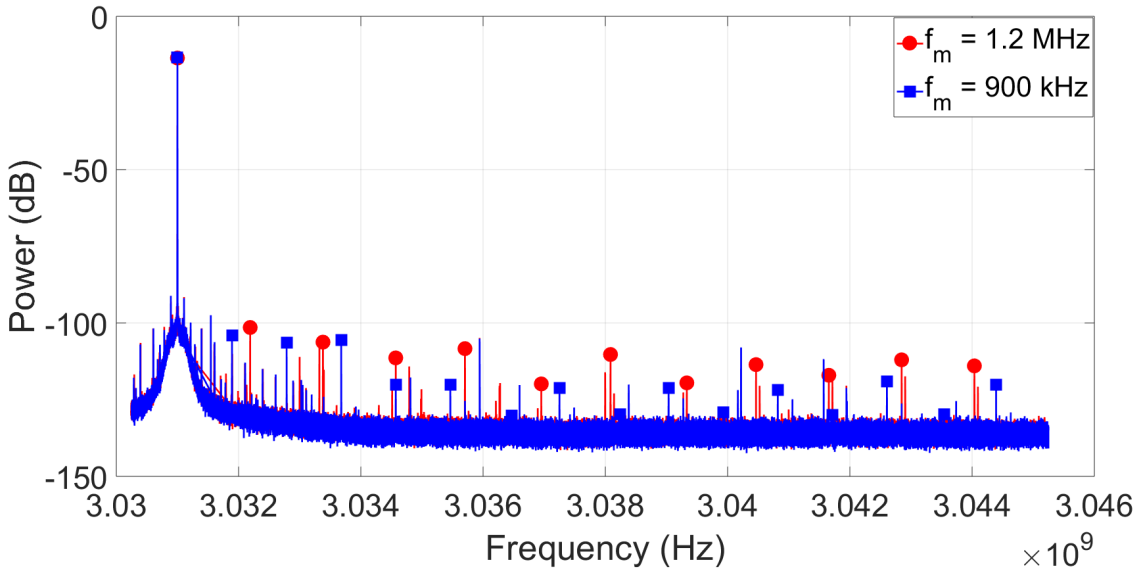


Figure 3.4: Measured backscatter power with $f_{carrier}$=3.031 GHz and f_m=900 kHz (blue), 1.2 MHz (red), respectively.

Note that these measurements were conducted in an indoor office environment, in the presence of measurement instruments, LCD monitors, mobile phones, WiFi routers, etc. that all create interference at various frequencies. While this can be a problem for measurements using the traditional electromagnetic side-channel, where some of the interference may be in the same frequency bands in which the chip produces side-channel emanations, with the backscattering side-channel such interference can be avoided by selecting $f_{carrier}$ such that no strong interference is present in a wide frequency band around it. Finally, please note that signal injected into the board is well below levels that may cause faults (whether transient or permanent) on the FPGA chip or elsewhere on the board.

3.3 The Advantages of Backscattering Side-Channel

Unlike other analog side-channels such as electromagnetic emanation (EM) and power, which are a consequence of current-flow changes inside the chip, backscattering side-

channel is an impedance-based side-channel that is the consequence of impedance switching activities inside the chip. These channels can be created by propagating a continuous-wave signal toward the chip. The transistor switching activities cause changes in the chip impedance, which modifies the radar cross-section (RCS) of the circuit. This RCS change modulates the signal that is backscattered (reflected) from the chip, which creates an impedance-based backscattering side-channel. If hardware Trojan is added to a circuit, it changes the impedance of the circuit even if the Trojan is not activated. The changes will be reflected in the backscattered signal, which is beneficial to the detection of hardware Trojan.

The backscattering side-channel has several advantages compared to other side-channels such as EM and power. These advantages can be listed as follows:

- *High bandwidth:* This provides the capability of detecting small and fast switching Trojan activities.

- *Signal strength not limited by leakage from devices:* One characteristic that sets the backscattering side-channel aside from others is that its signal strength can be improved by increasing the carrier's input power. As a result, the backscattering side-channel can still work when there is very little leakage from devices.

- *Adaptable frequency:* By changing the carrier frequency, we can change the working frequency of the backscattering side-channel. This helps to increase the signal-to-noise ratio by shifting the frequency to avoid interrupts that might distract the changes caused by HT activities.

3.4 Hardware Trojan Detection Using The New Backscattering Side-Channel

Switching in digital circuits causes internal impedances to vary, which causes changes in the circuit's radar cross-section (RCS), and thus modulates the carrier wave that is backscattered by the circuit. This new side-channel is impedance-based, so it can be beneficial to

detection of HTs because the HTs added circuitry, and also the additional connections attached to existing circuitry, result in modifications to the chip's RCS and in how that RCS changes as the on-chip circuits switch. Note that although the HT's trigger tends to be small, it exhibits switching activity as its logic reacts to inputs from the original circuitry, and it adds connections to the chip's original circuitry to obtain those inputs.

Most digital logic circuits are synchronous, so the overall switching pattern follows the clock cycle. Furthermore, the clock cycle usually accommodates switching delays along entire paths of logic gates, which means that the impedance changes of individual gates occur abruptly at some point in the clock cycle, i.e., they have a square-wave-like waveform. This implies that the backscattered signal will contain side-band components for several harmonics of the circuit's clock frequency f_C. These side-band components will be at $f_{carrier} \pm f_C$, $f_{carrier} \pm 2f_C$, $f_{carrier} \pm 3f_C$, etc., and the components at $f_{carrier} \pm f_C$ (that correspond to the first harmonic of the clock frequency) will mostly follow the overall RCS change during a cycle, while the components for the remaining harmonics will be influenced by the rapidity (rise/fall times) and timing of the impedance changes within the clock cycle.

Therefore, The detection of HTs using the backscattering side-channel will rely on measuring the amplitude of the backscattered signal at $f_{carrier} \pm f_C$, $f_{carrier} \pm 2 * f_C$, ..., $f_{carrier} \pm m * f_C$, i.e. the side-bands for the first m harmonics of the clock frequency. Only the amplitude is used (i.e. the signal's phase and other properties are ignored), mainly because the amplitude at some desired frequency is relatively easy to measure, whereas the phase and other properties require much more sophisticated tuning, phase tracking, etc. Furthermore, please note that each clock harmonic produces two side-band components that have the same amplitude, so the measurement can be made more efficient by only measuring m points to the left, or m points to the right, of $f_{carrier}$. In this work, the author measures points to the right of the carrier, i.e. $f_{carrier} + f_C$, $f_{carrier} + 2f_C$, etc.

Let call the m amplitudes measured for a given circuit a *trace*, and each trace char-

acterizes the circuit's overall amount, timing, and duration of impedance-change activity during a clock cycle. Intuitively, HTs can then be detected by first collecting training traces, using one or more ICs that are known to be HT-free, and then HT detection on other ICs would consist of collecting their traces and checking if they are too different from the traces learned in training.

However, the amplitude of a received signal declines rapidly with distance. The measurements are performed close to the chip, so even small variations in positioning of the probes create significant amplitude changes, and would result in numerous false positives when training and detection are not using identical probe positioning (which is very hard to achieve in practice).

Fortunately, the distance affects all of the points in a trace similarly, i.e. distance attenuates all amplitudes in the trace by the same multiplicative factor. Therefore, rather than using amplitudes for trace comparisons, The author uses amplitude ratios, i.e. amplitude of a harmonic divided by the amplitude of the previous harmonic[1], which cancels out the trace's distance-dependent attenuation factor. The resulting $m-1$ amplitude ratios are then used for comparing traces.

To illustrate amplitude ratios and how they are affected by differences in the tests circuit, Fig. 3.5 shows the statistics (mean and standard-deviation error bars) of each amplitude-ratio point, for a genuine AES circuit [31], and for the same AES circuit to which the T1800 Trojan from TrustHub [102] has been added but remains inactive throughout the measurement. In this experiment the carrier frequency is $f_{carrier}$=3.031 GHz, the AES circuit is clocked at f_C=20 MHz, and amplitudes for $m = 35$ right-side-band harmonics are measured to obtain the 34 amplitude ratios shown in Fig. 3.5.

It is observed that different amplitude-ratio points for the same trace vary significantly, from -30dB to 35dB in Fig. 3.5, and that different measurements for the same amplitude-

[1]Measurement of signal amplitude are often expressed in decibels, i.e. on a logarithmic scale, and for these measurements subtraction of logarithmic-scale amplitude values yields the logarithmic-scale value for the amplitude ratio

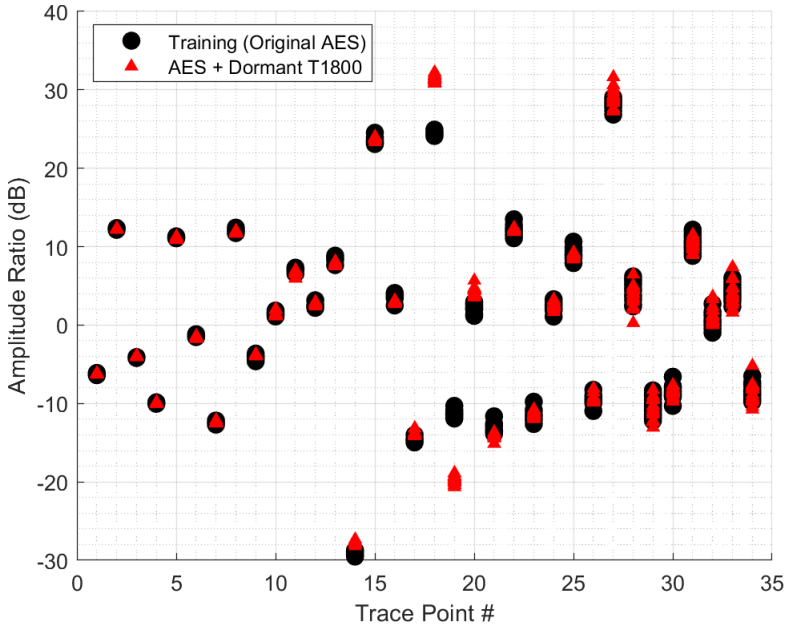


Figure 3.5: Amplitude ratios for HT-free and HT-afflicted AES.

ratio point tend to vary much less than that, making these differences difficult to see in Fig. 3.5, except for the very large differences between the HT-free and HT-afflicted design at the 18th and 19th amplitude ratio. This indicates that the impedance change is very small and the differences can be observed only at higher harmonics of the clock.

To more clearly show the differences at other harmonic-ratio points, Fig. 3.6 shows amplitude-ratio points that have been normalized to the mean amplitude ratio for the genuine AES circuit, i.e. for each amplitude ratio the logarithmic-scale points are shifted such that the genuine AES circuit's mean amplitude ratio becomes zero. It can now be observed that, in addition to the 18th and 19th point, which exhibit very large differences between the HT-free and the HT-afflicted measurements, the two circuits differ significantly in a number of other points, e.g. measurements for the two circuits are fully separable using the 14th point or the 20th point, and numerous other points have very little overlap between the HT-free and the HT-afflicted sets of measurements.

From Fig. 3.6, it can also be observed that the variance among measurements for the same design tends to increase with the index of the amplitude-ratio point, i.e., for points that correspond to higher harmonics. The primary cause of this increased variance is that higher

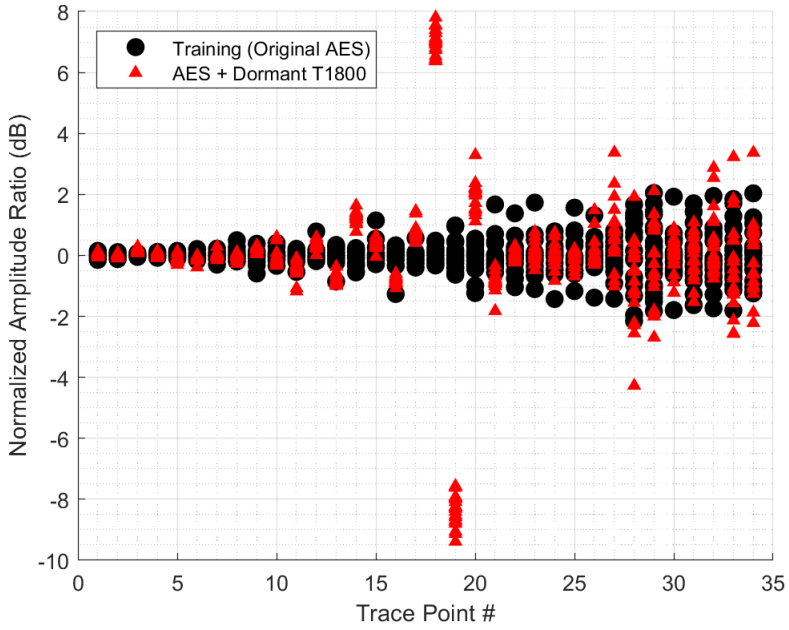


Figure 3.6: Amplitude ratios for HT-free and HT-afflicted AES, with each point normalized to the mean of its HT-free measurements.

harmonics of the signal tend to have lower amplitude, which makes their measurement less resilient to noise. Another factor that helps explain this increase in variance among higher harmonics is that they are affected by very small differences in timing of impedance changes during the clock cycle, and factors such as temperature and power supply voltage fluctuation can create small changes in the switching speed of the gates, and thus in the timing of the resulting impedance changes.

Regardless of the reason for the increasing variance among measurements of higher harmonics, the fact that the variance does increase is an important motivation for using an impedance-based side-channel rather than one created by bursts of current. Specifically, for each gate that switches, the impedance change persists for the rest of the cycle, while the burst of current is very brief in duration. This means that the impedance-change contributes to lower frequencies than the current-burst signal. When activity from cycle to cycle is repetitive, the spectrum of the signal's within-a-cycle waveform is projected onto the harmonics of the clock frequency, so gate-switching activity tends to affect lower harmonics of the clock frequency in impedance-based than in current-burst based side-channels. As

lower harmonics tend to have less variance from measurement to measurement, impedance-based side-channels can be expected to perform better for HT detection than current-burst based side-channels, and the results in Section 4.4.2 in Chapter 4 confirm that.

3.5 Hardware Trojan Detection Algorithm

Here, we present the very first HT detection prototype using backscattering side-channel. Please note that this prototype assumes a "golden" IC (known to be HT-free) can be used as a reference for training of the HT-detection mechanism. While we realize that this assumption is often unrealistic for practical deployments of HT detection, we evaluate HT detection with this assumption because it allows for a fair comparison with another side-channel (the EM side-channel). Removing the golden-reference assumption would make the results heavily dependent on the accuracy of the model and the simulator that generates the reference signals, and different side-channels would require different models/simulators that would be hard to equalize in accuracy/quality. Thus, we choose to evaluate the new backscattering side-channel, and to compare it to the EM and power side-channel, under the same assumptions/conditions, in order to demonstrate the advantages of this new side-channel, namely that it can detect much smaller circuit modifications, is less susceptible to manufacturing variability, and can detect dormant HTs. In addition, the assumption makes it easier to evaluate how changes in the size and position of HT's trigger and payload affect the detectability of HTs.

This HT detection prototype has two phases: *training*, where a circuit that is known to be HT-free is characterized, and *detection*, where an unknown circuit is classified into one of the two categories – HT-free or HT-afflicted, according to how much its measurements deviate from the statistics learned in training.

3.5.1 Training

Fig. 3.7 details the training for the prototype implementation of backscattering-based HT detection. This training consists of measuring K times the signal backscattered from an IC known to be HT-free, each time collecting the m amplitudes at frequencies that correspond to the lowest m harmonics of the IC's clock frequency in the side-band of the received backscattered signal. The $m-1$ amplitude ratios are then computed from these amplitudes. Next, for each of the $m-1$ amplitude ratios, the mean and standard deviation across the M measurements are computed, and the detection threshold for HT detection is computed as the sum of the $m-1$ standard deviations.

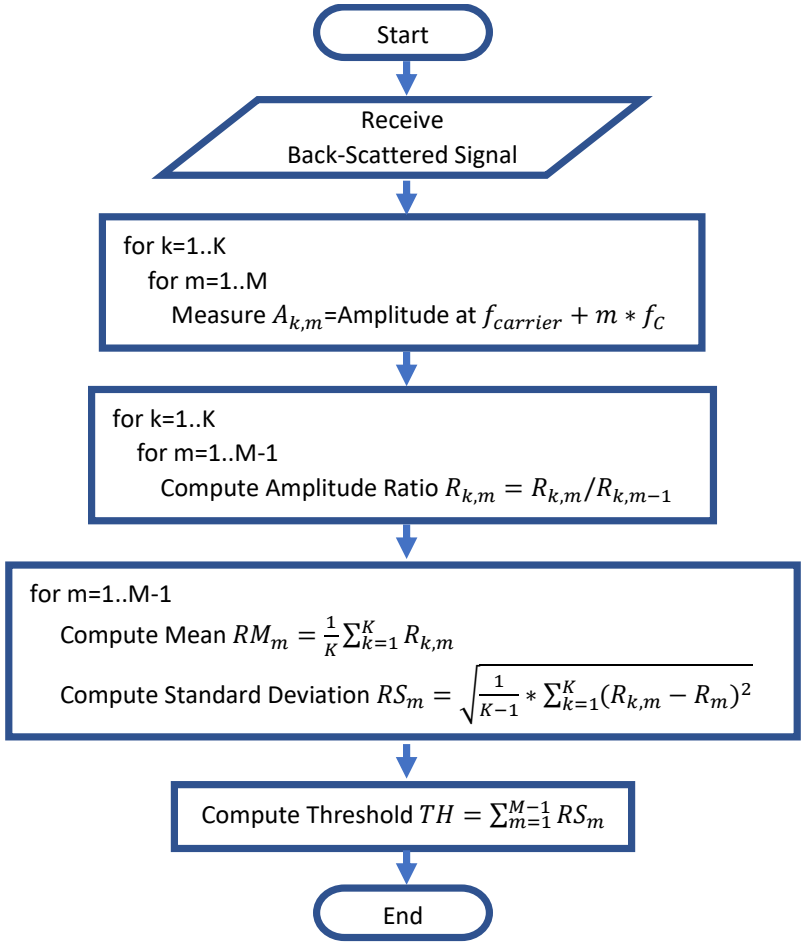


Figure 3.7: Training algorithm.

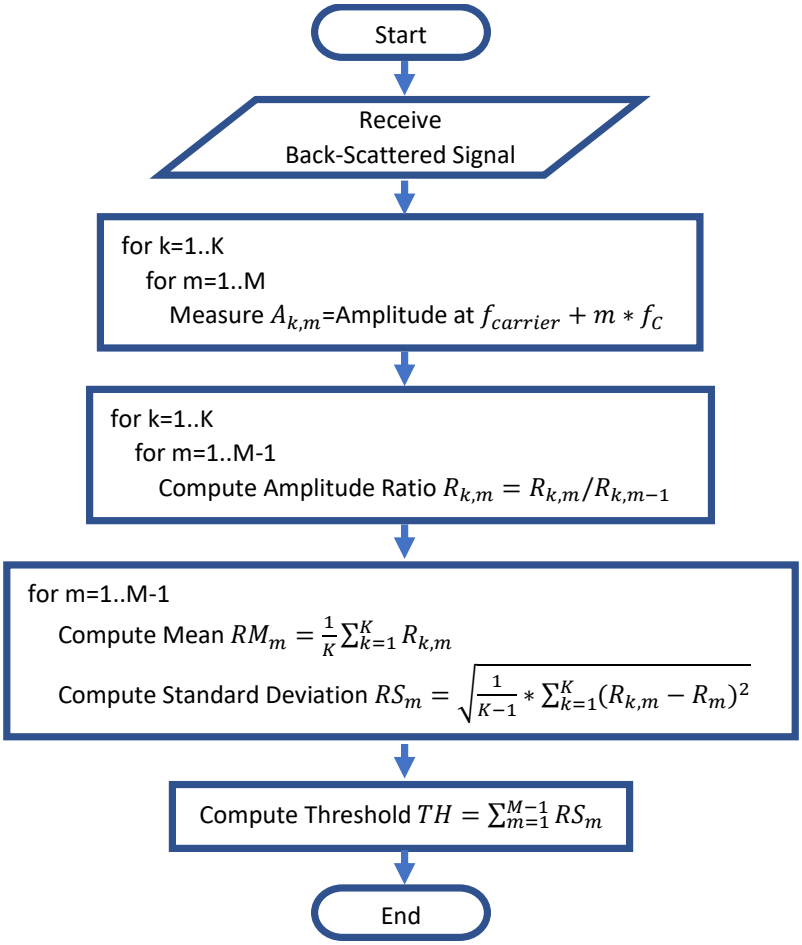


Figure 3.8: Test algorithm.

3.5.2 Detection

Figure 3.8 details how the prototype implementation of backscattering detection decides whether to classify an IC as HT-free of HT-afflicted. First, a single measurement is obtained of the m amplitudes that correspond to the lowest m harmonics of the IC's clock frequency in the side-band of the signal that is backscattered from the IC under test, and $m - 1$ amplitude ratios are computed from these amplitudes.

Next, for each of the $m-1$ amplitude ratios, we compute how much it deviates from the corresponding mean computed during training. This deviation is computed as the absolute value of the difference, and intuitively it measures how much that amplitude ratio differs from what would be expected from an HT-free IC. Finally, this sum of these deviations is compared to the sum of standard deviations from training. Intuitively, the sum of the differ-

ences for the IC under test is a measure of how much its overall backscattering "signature" differs from what would be expected from an HT-free IC, and the sum of standard deviations from training corresponds to how much an individual measurement of an HT-free IC can be expected to differ from the average of HT-free measurements. The IC under test is labeled as HT-free if its sum of amplitude-ratio deviations is lower than this detection threshold (sum of standard deviations from training).

3.6 Experimental Setup

3.6.1 Backscattering Side-Channel Measurement Setup

Figure 3.9 shows the measurement setup that we use to evaluate the performance of the proposed prototype backscattering-based HT-detection. The carrier signal is a sinusoid at $f_{carrier}$=3.031 GHz produced by an Agilent MXG N5183A signal generator and transmitted toward the FPGA chip using an Aaronia E1 electric-field near-field probe. To select $f_{carrier}$, we have measured signal strength at the frequency of the reflected carrier signal (the signal we were injecting into the board), the first several harmonics of the modulated FPGA board clock (e.g. 50 MHz away from the carrier), and of the noise floor of the instrument using AARONIA Near Field Probes (0 to 10 GHz). We have found that the side-band signal for the first harmonic of the board's clock is strongest when $f_{carrier}$ is around 3 GHz, but we have also found that traditional EM emanations create interference at frequencies that are multiples of the board's clock frequency (50MHz). Thus, we choose $f_{carrier}$=3.031 GHz, a frequency close to 3GHz that avoids interference from the board's traditional EM emanation. The device-under-test (DuT) is the FPGA chip on the Altera DE0-CV board, and it is positioned using a right-angle ruler so that different DE0-CV boards can be tested using approximately the same position of probes. The backscattered signal is received with an Aaronia H2 magnetic field near-field probe, and this signal is pre-amplified using an EMC PBS2 low-noise amplifier and then the signal amplitudes at desired frequencies are measured using an Agilent MXA N9020A Vector Signal Analyzer.

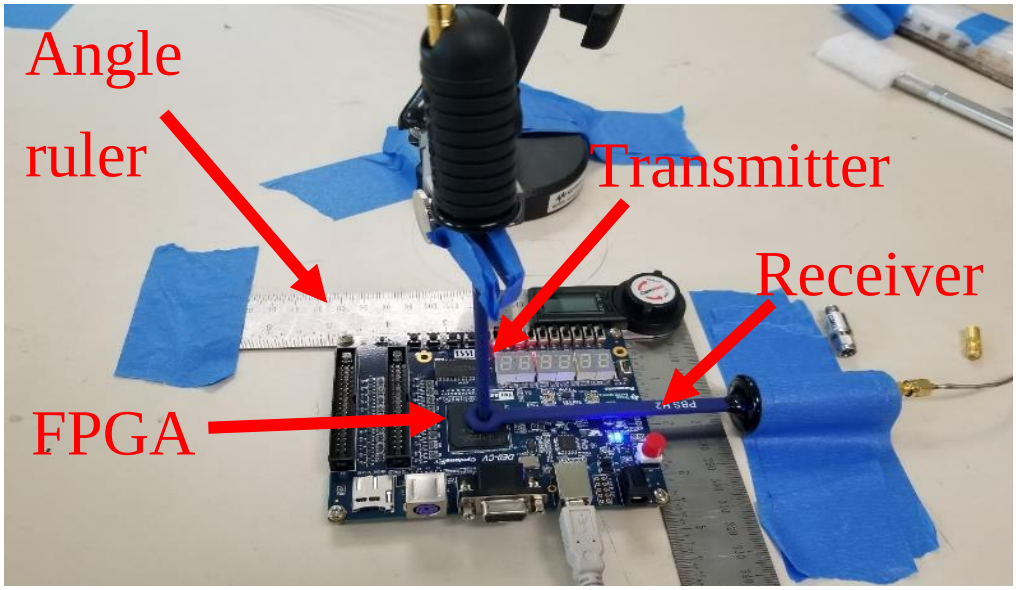


Figure 3.9: Measurement setup for hardware Trojan detection using back-scattering side-channel.

3.6.2 Training and Testing Circuit Designs

All circuits used in our experiments are implemented on a Field Programmable Gate Array (FPGA), which allows rapid experimentation by changing the circuit and/or its physical placement and routing, unlike hard-wired ASIC designs that would require fabrication for each layout variant of each circuit. The specific FPGA board we use is the Altera DE0-CV board, and within it, the IC on which our backscattering measurement setup focuses is the Altera 5CEBA4F23C7N, an FPGA in Altera‘s Cyclone V device family.

For our HT detection experiments, we use AES-T1800, AES-T1600, and AES-T1100 hardware Trojan benchmarks from TrustHub [102]. For all three of these HTs, the original HT-free design is an AES-128 cryptographic processor, which uses an 11-stage pipeline to perform the 10 stages of AES encryption on 128-bit block. Since numerous HTs in the TrustHub repository are similar to each other, we selected these three HT benchmarks because they exhibit different approaches for their triggers and payloads.

- T1800: The payload in this HT is a cyclic shift register that, upon activation, continuously shifts to increase power drain consumption, which would be a serious problem

for small battery-powered or energy-harvesting devices in e.g., medical implants. The HT's trigger circuit consists of combinatorial logic that monitors the 128-bit input of the AES circuit, looking for a specific 128-bit plaintext value, and the occurrence of that 128-bit value at the input activates the payload. The size of T1800's trigger circuit is 0.27% of the original AES circuit, and the size of its payload is 1.51% of the size of the AES circuit. Because this HT's trigger and payload can be resized easily, we use this HT to study how our HT detection is affected by HT size and physical location.

- T1600: The payload in this HT creates activity on an otherwise-unused pin to generate an RF signal that leaks the key of the AES circuit. The HT's trigger circuit consists of sequential logic which activates the payload when a predefined *sequence* of values is detected at input of the AES circuit. The size of T1600's trigger circuit is 0.28% of the size of the original AES circuit, while the size of its payload is 1.76% of the size of the original AES circuit.

- T1100: The payload of this HT modulates its activity using a spread-spectrum technique to create a power consumption pattern that leaks the AES key. The trigger is a (sequential) circuit that looks for a predefined sequence of values at the input of the AES circuit to activate the payload. The size of T1800's trigger circuit is 0.28% of the size of the original AES circuit, while the size of its payload is 1.61% of the size of the AES circuit.

A key challenge we faced when implementing the HT-afflicted circuits was that these HTs are specified at the register-transfer level, as modifications to the original AES circuit's Verilog HDL source code. If the modified source code is subjected to the normal compilation, placement, and routing, we found that the addition of the HT causes the EDA tool to change the placement and routing of most logic elements in the overall circuit, and this extensive change makes the modification very easy to detect regardless of the HT's

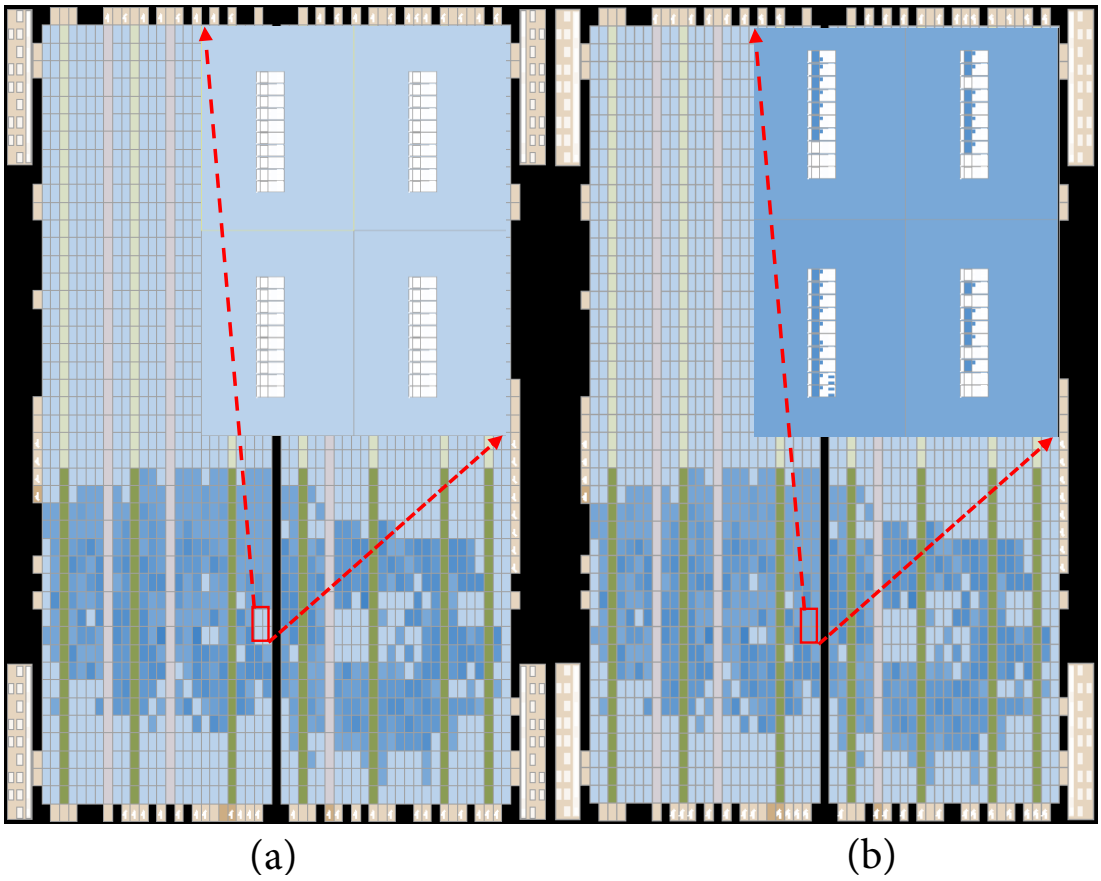


Figure 3.10: (a) Genuine AES circuit (b) Hardware Trojan infected AES circuit.

actual size and activity. The next approach we tried was to compile the AES circuit using the normal compilation, placement, and routing, and then for each HT-afflicted design we used the ECO (Engineering Change Order) tool in Altera's Quartus II suite to add the HT's circuitry while leaving unchanged the placement of logic elements (and the routing of their connections) that belong to the original AES circuit. However, we found that this approach makes it very hard to place the HT's logic elements close to the inputs of the original AES circuit, and (as will be demonstrated in Section 3.7.3), the HT is easier to detect when its trigger is placed away from where it is connected to the original circuit. To make the HTs more stealthy, we instead compile, place, and route the *HT-afflicted* circuit, then create the HT-free circuit by removing (using the ECO tool) the HT's logic elements and their connections. To illustrate this, the placement of the HT-free circuit and the T1800-afflicted circuit are shown in Fig. 3.10, with a zoom-in to show the details where the HT's logic

elements are placed.

Finally, for HT detection, the circuit must be supplied with inputs during the evaluation. Since we evaluate our HT detection approach in the dormant-HT scenario, any input sequence that causes logic gates in the original AES circuit to change state can be used, so each cycle we simply flip all of the AES circuit's input bits, as shown in Fig. 3.11.[2]

```
always @ (posedge clk or posedge rst)
begin
  if (rst == 1'b1) begin
     cnt = 1'b0 ;
  end else begin
     if (cnt == 1'b1) begin
        cnt = 1'b0 ;
     end else begin
        cnt = cnt + 1'b1 ;
     end
  end
end

always @ (posedge clk or posedge rst)
begin
  if (rst == 1'b1) begin
     r_state <= 128'h55555555_55555555_55555555_55555555 ;
  end
  else begin
        case (cnt)
           1'b0: r_state = 128'h55555555_55555555_55555555_55555555 ;
           1'b1: r_state = 128'hAAAAAAAA_AAAAAAAA_AAAAAAAA_AAAAAAAA ;
        endcase
  end
end
```

Figure 3.11: Feeding inputs to the AES circuit.

3.7 Evaluation

3.7.1 Detection of Dormant vs. Active Hardware Trojan Using the Backscattering Side-Channel

Because it is very difficult to activate an HT without a priori knowledge of its trigger conditions, it is highly desirable for an HT detection scheme to provide accurate detection of *dormant* HTs, i.e., to detect HTs whose payload is never activated while it is characterized by the HT detection scheme. However, a dormant HT is typically more difficult to detect compared to an activated HT. For side-channel-based detection methods, in particular, the switching activity in the activated payload, and/or the changes it creates in the switching

[2]Note that hexadecimal 3 and C correspond to binary 0011 and 1100, while hexadecimal A and 5 correspond to 1010 and 0101, respectively. Thus the inputs we feed to the AES circuit simply toggle each of the input bits, while avoiding all-ones and all-zeros patterns.

activity of the original circuit, have more impact on the side-channel signal than an inert payload (no switching activity in the payload and no changes to the original circuit's functionality).

Figure 3.12 compares the normalized amplitude ratios for an HT-free AES design and for the same AES design (and layout) to which the AES-T1800 Trojan has been added. Two separate sets of 20 measurements are shown for the HT-free design, one that is used for training and one that is used to detect false positives when evaluating HT detection (on another DE0-CV board). For the HT-afflicted design, one set of 20 measurements is collected when the HT is dormant (its payload has not been activated), and another set of 20 measurements is collected with the same HT after its payload is activated. We can observe

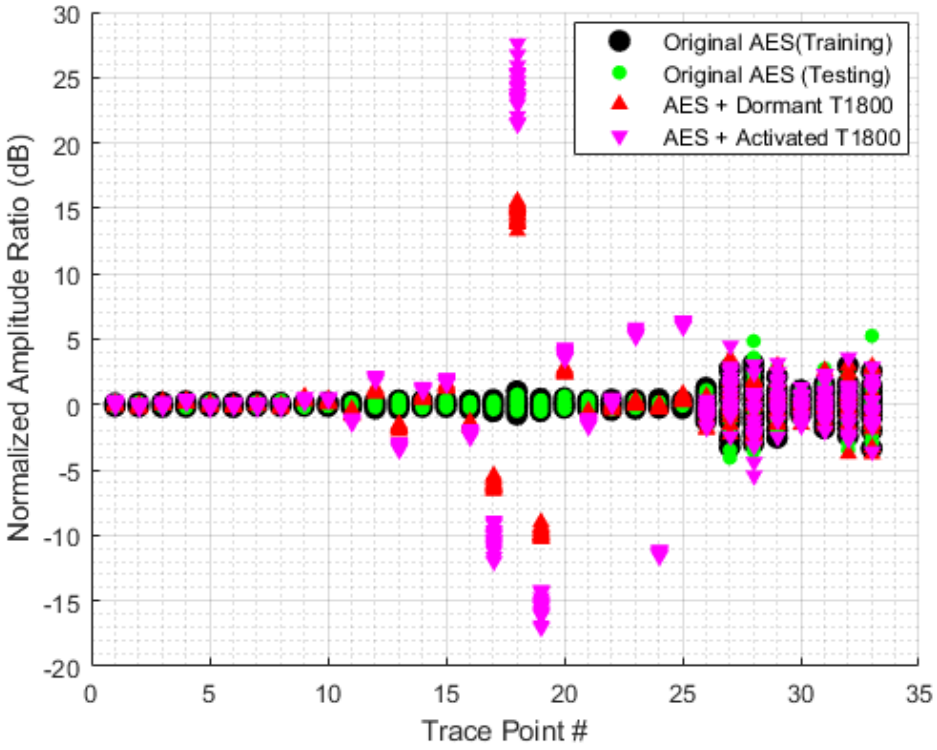


Figure 3.12: Normalized amplitude ratios for backscattering side-channel measurements.

that there are a number of trace points where both sets of HT-afflicted measurements deviate significantly from HT-free measurements, and that this deviation tends to be larger for measurements in which the HT has been activated. The higher deviation from HT-free mea-

surements seen for active-HT measurements agrees with the intuitive reasoning that an HT is easier to detect when active then when it is dormant. Even so, our backscattering-based HT detection prototype successfully reports the existence in each dormant-HT experiment (100% detection rate), while correctly reporting all 20 HT-free measurements as HT-free (no false positives).

3.7.2 Detection of Dormant Hardware Trojan with Cross-Training Using the Backscattering Side-Channel

Another important practical concern for HT detection is robustness to manufacturing variations and other differences between different physical instances of the same hardware design. Thus our evaluation focuses on detection of *dormant* HTs with *cross-training*, i.e., training for HT detection is performed on one hardware instance, and then HT detection is performed on others. We evaluate the effectiveness of our HT detection prototype by training it on one DE0-CV FPGA board with an HT-free AES circuit, then applying HT detection to several test subject circuits implemented on nine DE0-CV FPGA boards, none of which is the same as the one used for training. The test subject designs are: *Original AES*, *AES + Dormant T1800*, *AES + Dormant T1600*, *AES + Dormant T1100*.

For each measurement, the previously measured FPGA board is removed from the measurement setup, and then a different board is positioned using an angle ruler to model a realistic measurement scenario when each measurement uses a very similar but not identical relative position of the chip and the probes. Each test subject design is measured 20 times on each board, and each measurement is used for HT detection in isolation, i.e. for each test subject the detection makes 20 classification decisions (HT-free or HT-afflicted) on each of the 9 boards, resulting in a total of 720 decisions. Among these decisions, 180 were on the *Original AES* test subject, and in all 180 of them our prototype has correctly classified the design as HT-free, i.e., the HT detection prototype had no false-positive detections. In the remaining 3 sets of 180 decisions, each test subject design was HT-afflicted

(180 decisions with T1800, 180 decisions with T1600, and 180 with T1100), and in all of them our prototype has correctly classified the design as HT-afflicted, i.e. the HT detection prototype has detected the presence of an HT in each measurement in which an HT was present.

3.7.3 Impact of Trojan Trigger and Payload's Size and Position on Hardware Trojan Detection

Impact of Trojan Trigger and Payload's Size

To provide more insight into which factors influence our HT detection prototype's ability to detect dormant HTs, we perform experiments in which we reduce the size of the T1800 hardware Trojan's trigger and payload. The T1800 was chosen because it has the smallest trigger among the HTs we used in our experiments, and because both its payload and its trigger can be meaningfully resized.

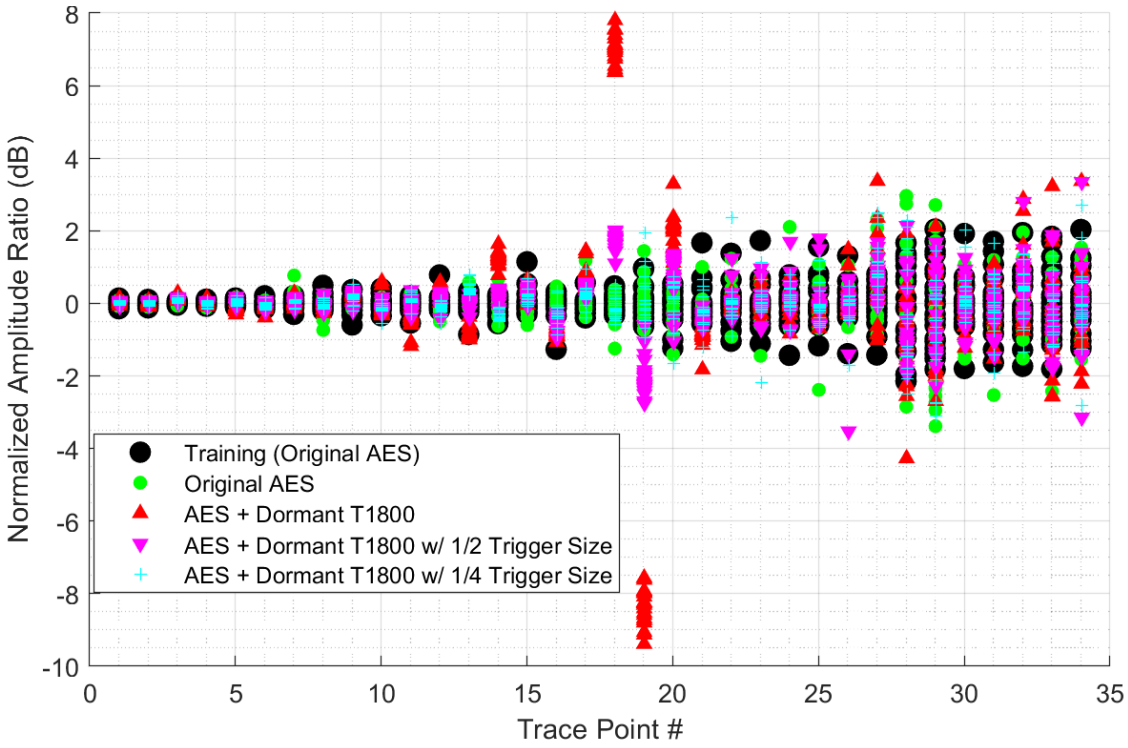


Figure 3.13: Normalized amplitude ratios for different sizes of T1800's trigger input.

The T1800 monitors the 128-bit data input of the AES-128 circuit, comparing it to a specific hard-wired 128-bit value, and it activates the payload when that 128-bit value is detected. In terms of logic elements (gates), the size of this 128-bit trigger is only 0.27% of the size of the original AES circuit, i.e. even this full-size trigger is much smaller than the AES circuit to which the HT has been added, and its activity (while the HT is dormant) is difficult to detect using existing side-channels. We implement reduced-trigger variants of this HT by monitoring only the 64 least significant bits (the "1/2 Trigger Size" variant, where the trigger circuit size is only 0.15% of the original AES circuit's size), and then only the 32 least significant bits (the "1/4 Trigger Size" variant, where the trigger circuit size is only 0.08% of the original AES circuit size). The normalized harmonic ratio traces for 20 measurements of each design, along with 40 HT-free measurements (20 for training and 20 for false-positives testing) are shown in Fig. 3.13. We observe that smaller trigger sizes result in trace points that are closer to HT-free ones, i.e., that trigger size directly impacts the side-channel-based separation between dormant-HT and HT-free circuits. These results match the intuition that the HT's influence on impedance changes should increase as more input bits are monitored by the HT's trigger, both because of the increased number of connections to the original circuit (which can change impedances "seen" by gates that belong to the original circuit) and because of the increased number of gates whose values can change (switching activity) within a cycle in the HT's trigger circuit itself.

The ROC curves for HT detection with different trigger sizes (Fig. 3.14) confirm that, while the HT with the original-size and even 1/2-size trigger can be detected in each measurement with no false positives, the detection accuracy suffers significantly as the HT's trigger is further reduced to 1/4 of the original size.

We perform additional experiments in which we keep the trigger at full size, but reduce the size of the payload to 50% and then 25%. Our dormant-HT measurement results for these variants are not noticeably different from each other (Fig. 3.15), which implies that the payload size has little impact on our HT detection. This agrees with our theoretical and

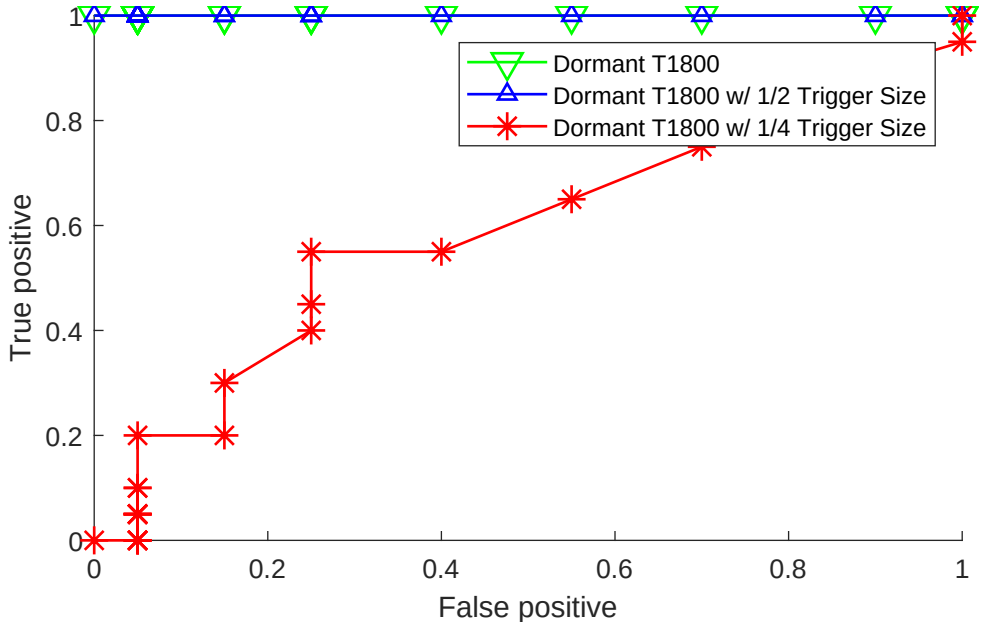


Figure 3.14: ROC curves for HT detection for different sizes of the HT's trigger circuit.

intuitive expectations: the payload in T1800 has little impact on the impedance changes during a clock cycle, as it has no switching activity (until activated), and has no connections to the gates in the original AES circuit (T1800's payload is designed to produce a lot of power-draining switching activity upon activation, not to change the functionality of the AES circuit).

Since the measurements of the full-trigger-and-reduced-payload variants of T1800 HT are very similar to the full-size T1800 HT, they provide the same ROC curves (complete detection without false positives) as the full-size T1800 HT, as shown in Fig. 3.14.

Impact of HT Trigger and Payload's Position

We next investigate how the backscattering-based HT detection is influenced by the physical location and routing of the HT's connection to the original circuit. For this, we start with the AES circuit with the T1800 HT, whose trigger logic was placed at Position 1 shown in Fig. 3.16 by the placement and routing tool very close to where its 128-bit input can be connected to the original AES circuit.

We then create a variant of this HT by moving the HT's trigger logic to Position 2,

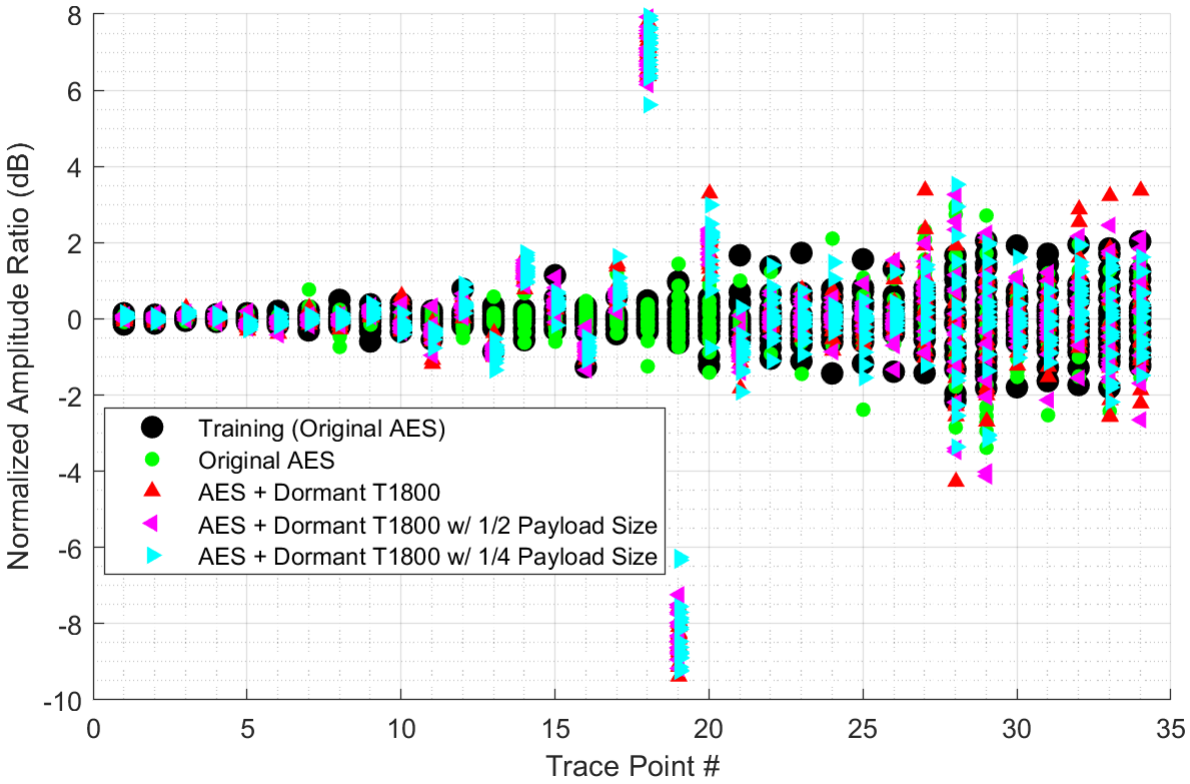


Figure 3.15: Normalized amplitude ratios for different sizes of T1800's (dormant) payload.

keeping the logic elements and the connections between them in the same position relative to each other, but making the trigger's 128 connections to the original AES circuit much longer. Another variant is similarly created by moving the HT's trigger logic to Position 3.

The dormant-HT measurement results for these three positions are shown in Fig. 3.17. We observe that, at many trace points, in terms of separation of HT-afflicted measurements HT-free ones, Position 2 is significantly more separated than Position 1, and Position 3 provides an additional small increase in separation. This means that HTs placed close to their connection points in the original circuit are more difficult to detect than HTs that require long connections. We also performed experiments in which the trigger part of the HT is kept in Position 1, while its payload was moved to Position 2 and then Position 3. Our results show that the payload position has little impact on the measurements, which is as expected given that, in our dormant-HT experiments, the 1-bit "activate" signal between the trigger and the payload never changes its value (it stays at 0, i.e. inactive), and that the

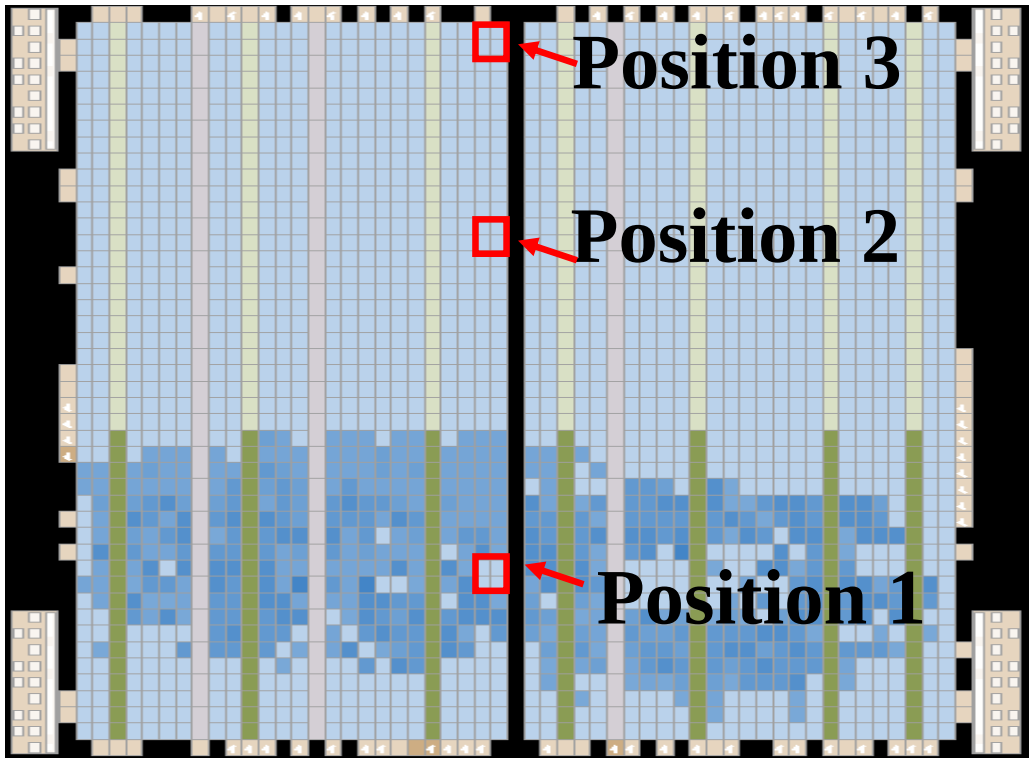


Figure 3.16: Changing the physical position off the HT's trigger logic.

payload has no switching activity.

3.8 Further Evaluation of Hardware Trojan Detection Using More Benchmarks

To further evaluate the effectiveness of our HT detection prototype, we implement two different circuits, RS232 and PIC16F84, each with three HTs, from TrustHub [102]. We use the same HT detection prototype described in Section 3.5 and the setup described in Section 3.6.

3.8.1 RS232 circuit

We use RS232-T500, RS232-T600, and RS232-T700 HT benchmarks from TrustHub [102]. For all three of these HTs, the original HT-free design is a RS232 micro-UART core consisting of a transmitter and a receiver. The transmitter takes input words (128-bit length) and serially outputs each word according to the RS232 standard, while the receiver takes a

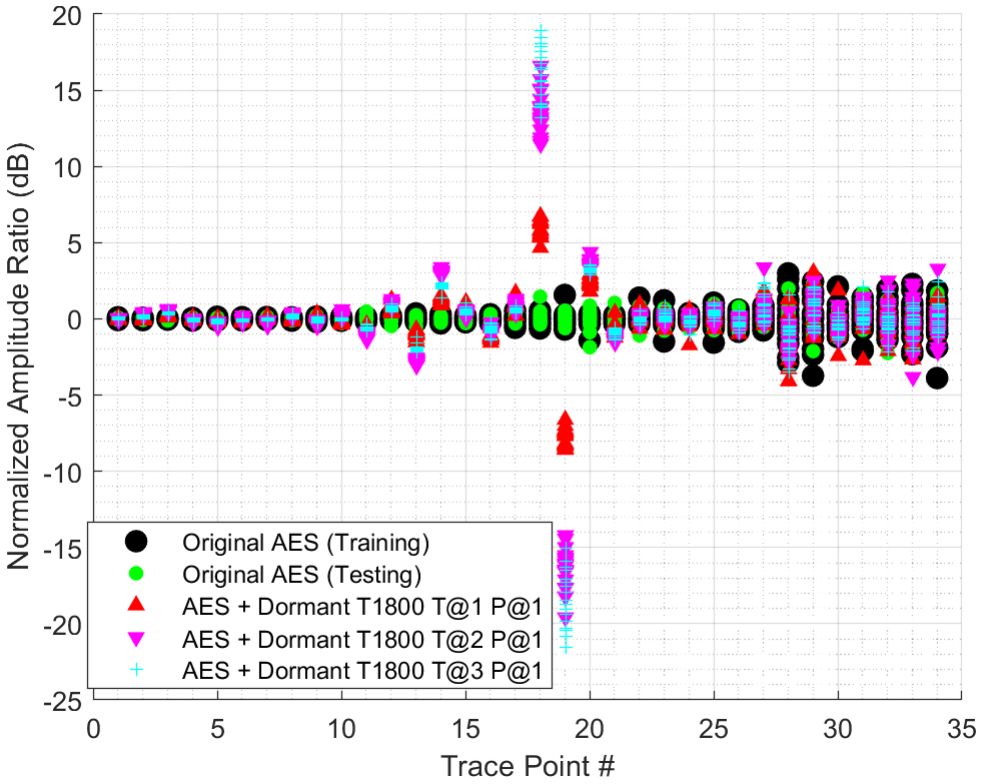


Figure 3.17: Normalized amplitude ratios for different locations of T1800's trigger logic.

serial input and output 128-bit words.

- RS232-T500: The payload in this HT is a circuit that, upon activation, causes the transmission to fail. The trigger is sequential circuit that increments its counter every clock cycle, and activates the payload activated when this counter reaches a certain value. The size of the trigger circuit is 1.67%, and the size of the payload circuit is 1.48% of the size of the RS232 circuit.

- RS232-T600: The payload in this HT is a circuit that, upon activation, makes the transmitter's "ready" signal become stuck-at-1, and changes specific bits in the transmitted data. The trigger is a sequential circuit that looks for a specific sequence of UART states to activate the payload. The size of the trigger circuit is 1.54%, and the size of the payload circuit is 1.52% of the size of the RS232 circuit.

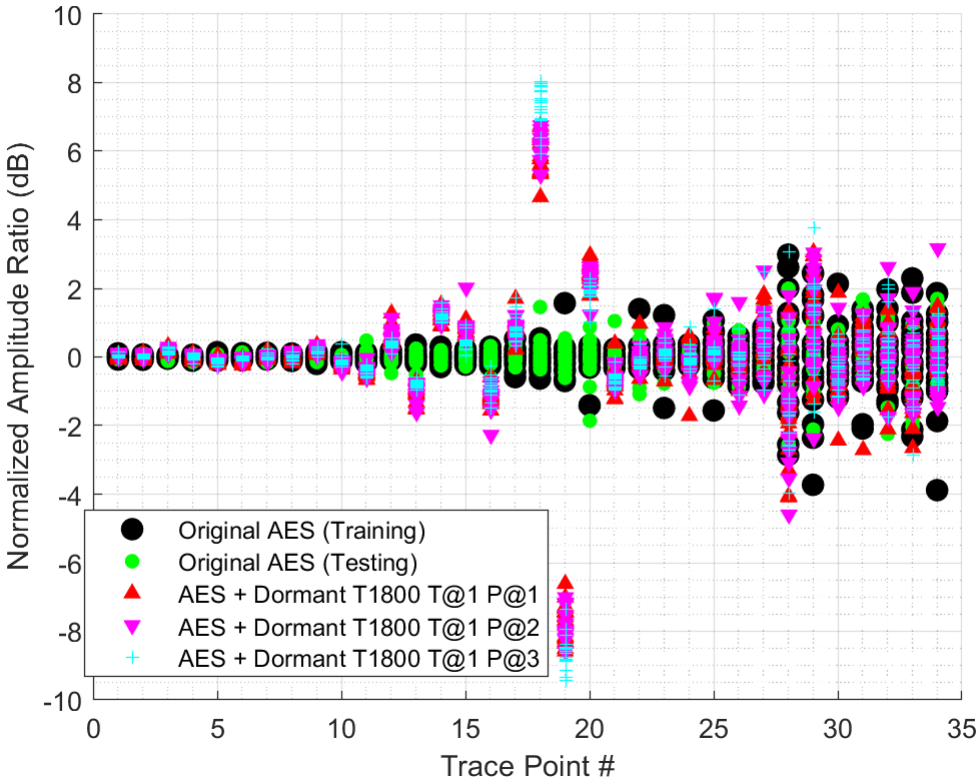


Figure 3.18: Normalized amplitude ratios for different locations of T1800's (dormant) payload.

- RS232-T700: The payload of this HT is a circuit that, upon activation, makes the transmitter's "finished" signal become stuck-at-0. The trigger is sequential circuit that looks for a predefined sequence of UART states to activate. The size of the trigger circuit is 1.54%, and the size of the payload circuit is 1.48% of the size of the RS232 circuit.

The results in Figs. 3.19 and 3.20 show the ratios of harmonics and ROC curve, respectively. The results show that we can detect each of these three Trojans with 100% accuracy and 0% false positives.

3.8.2 PIC16F84 circuit

We use PIC16F84-T100, PIC16F84-T200, and PIC16F84-T400 hardware Trojan benchmarks from TrustHub [102]. For all three HTs, the original HT-free design is PIC16F84

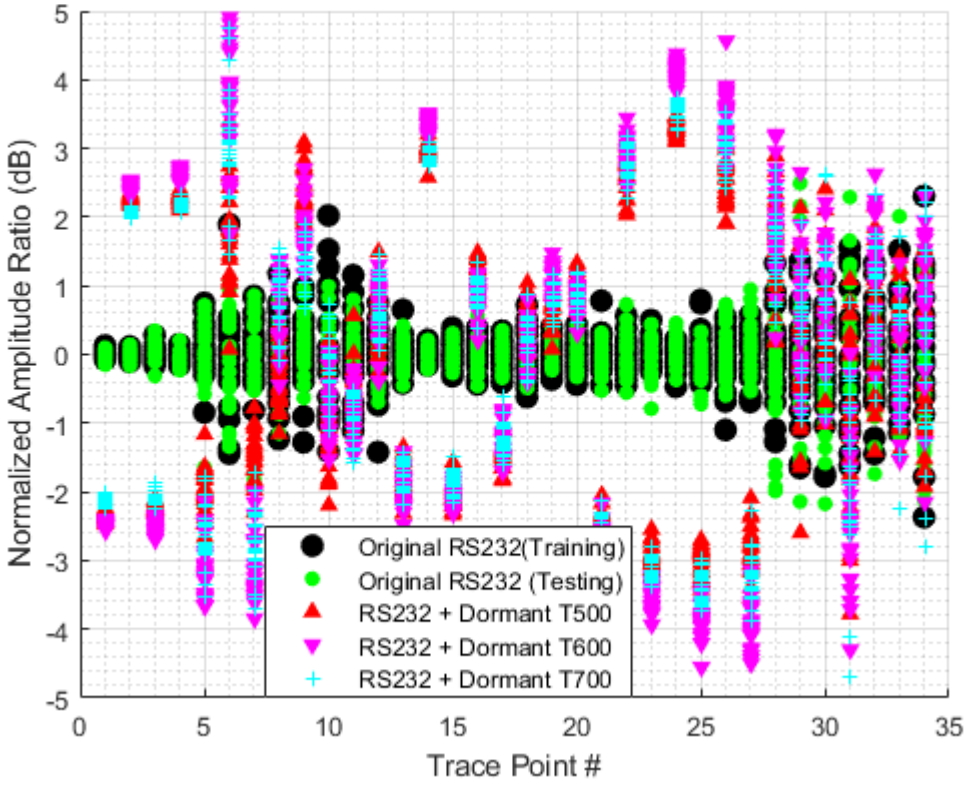


Figure 3.19: Normalized amplitude ratios for different HTs in the RS232 circuit.

circuit, a RISC micro-controller whose functions and instruction set are very similar to those of the Microchip 16F84 chip.

- PIC16F84-T100: Once activated by its (sequential) trigger circuit, the payload changes the address to PIC16F84's program memory (causing denial of service). The size of the trigger circuit is 1.34%, while the size of the payload circuit is 1.81% of the size of the PIC16F84 circuit.

- PIC16F84-T200: Once activated by its (sequential) trigger circuit, the payload in this HT replaces the instruction register with a sleep command (causing denial of service). The size of the trigger circuit is 1.35%, and the size of the payload circuit is 1.93% of the size of the PIC16F84 circuit.

- PIC16F84-T400: Once activated by its (sequential) trigger circuit, the payload of this HT changes the address lines to the external EEPROM to 0 (causing denial of

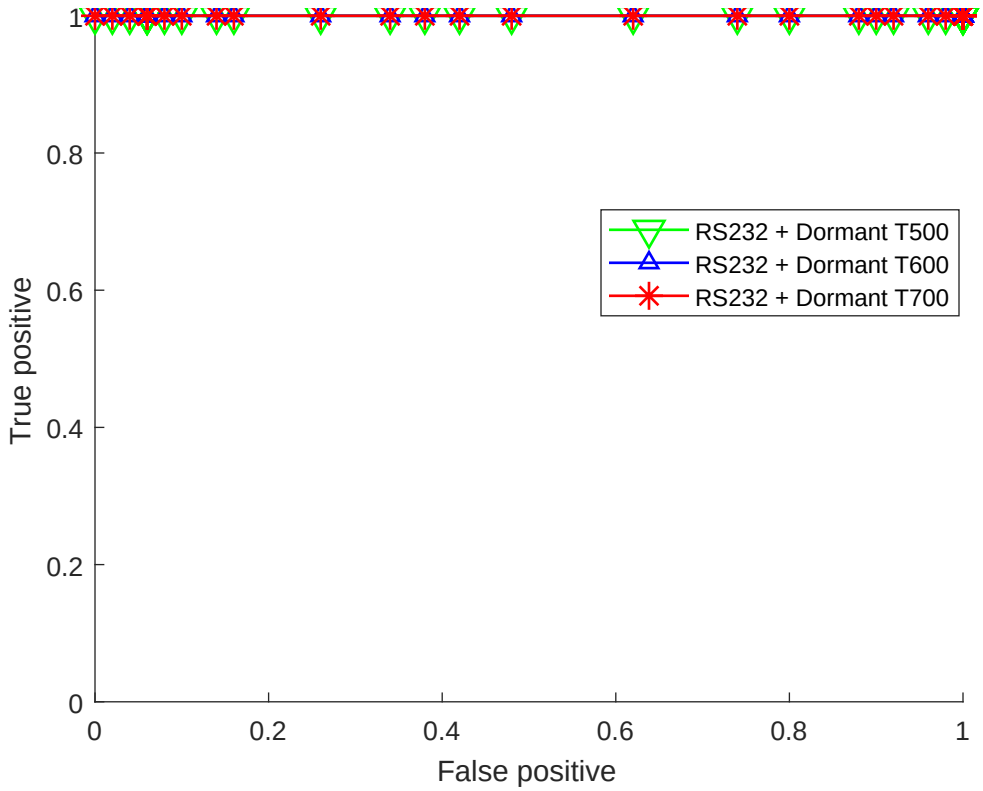


Figure 3.20: ROC curves for detection of HTs in the RS232 circuit.

service). The size of the trigger circuit is 1.35%, while the size of the payload circuit is 1.75% of the size of the PIC16F84 circuit.

The results in Figs. 3.21 and 3.22 show the ratios of harmonics and ROC curve, respectively. The results show that we can detect each of these three Trojans with 100% accuracy and 0% false positives.

3.8.3 Trigger Size Experiment

As discussed in Section 3.7.3, trigger size has a significant effect on dormant-HT detectability. We chose RS232-T500 for this experiment because its trigger consists of monitoring the executed instruction stream, counting occurrences of a specific instruction until a threshold value is reached (and then activating the payload). The counter's size can be changed with-

out affecting/changing the overall functionality of the HT[3]. Our reduced-trigger variants of this Trojan by reducing the number of bit of the counter. We have the following design:

- RS232 + Dormant T500: The size of the trigger is 1.67% of the size of the original RS232 circuit.
- RS232 + Dormant T500 w/ 1/2 Trigger: The size of the trigger circuit is 1% of the size of the original RS232 circuit.
- RS232 + Dormant T500 w/ 1/4 Trigger: The size of the trigger circuit is 0.67% of the size of the original RS232 circuit.
- RS232 + Dormant T500 w/ 1/8 Trigger: The size of the trigger circuit is 0.33%, of the size of the original RS232 circuit.

For all four of these variants, the payload circuit remains unchanged, and its size is 1.48% of the original RS232 circuit's size.

The results in Figs. 3.23 and 3.24 show that the smaller the trigger is, the harder it is to detect the Trojan, which agrees with our previous results for AES-based HTs.

3.9 Conclusions

This chapter describes a new physical side-channel, i.e., the backscattering side-channel, that is created by transmitting a signal toward the IC, where the internal impedance changes caused by on-chip switching activity modulate the signal that is backscattered (reflected) from the IC. To demonstrate how this new side-channel can be used to detect small changes in circuit impedances, we propose a new method for non-destructively detecting HTs from outside of the chip. To our knowledge, this is the first off-chip side-channel technique capable of detecting *inactive* HTs while tolerating variations that exist across hardware

[3]However, the reduced counter size requires the threshold to be reduced, thus activating the payload sooner and risking detection of the HT during functional and burn-in tests.

instances. Also, to our knowledge, backscattering has never before been used as a side-channel signal to infer information about the operation of electronic circuitry, even though backscattering has been used extensively for RFID tags and other short-range communications [30].

We experimentally confirm, using measurements on one physical instance for training and nine other physical instances for testing, that the new side-channel, when combined with an HT detection method, allows detection of a *dormant* HT in 100% of the HT-afflicted measurements for a number of different HTs, while producing no false positives in HT-free measurements. Furthermore, additional experiments are conducted to compare the backscattering-based detection to one that uses the traditional EM-emanation-based side-channel. These results show that backscattering-based detection outperforms the EM side-channel, confirm that dormant HTs are much more difficult for detection than HTs that have been activated, and show how detection is affected by changing the HT's size and physical location on the IC.

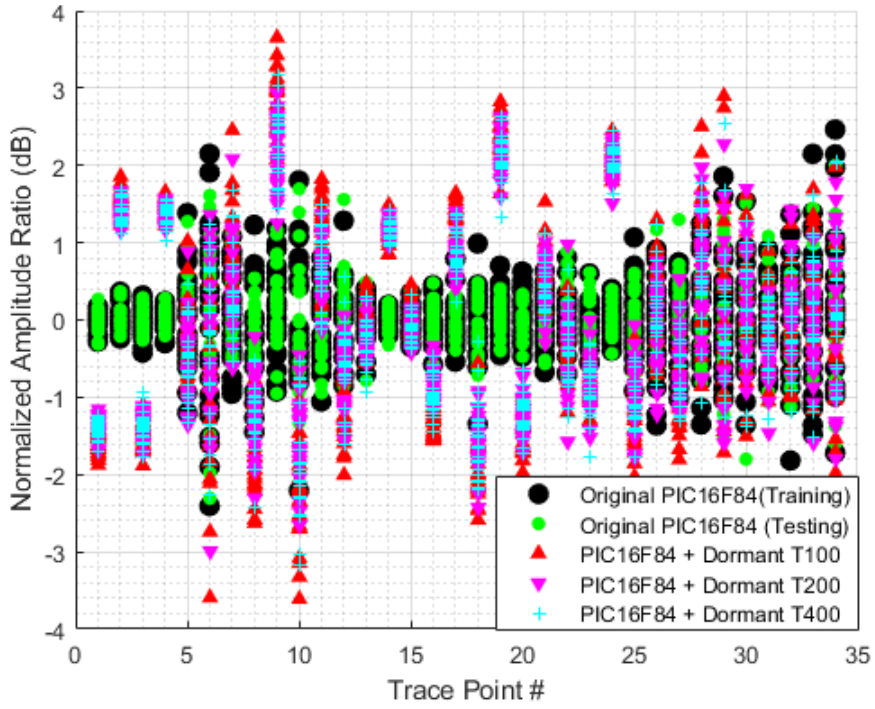


Figure 3.21: Normalized amplitude ratios for different Trojans on PIC16F84 circuit.

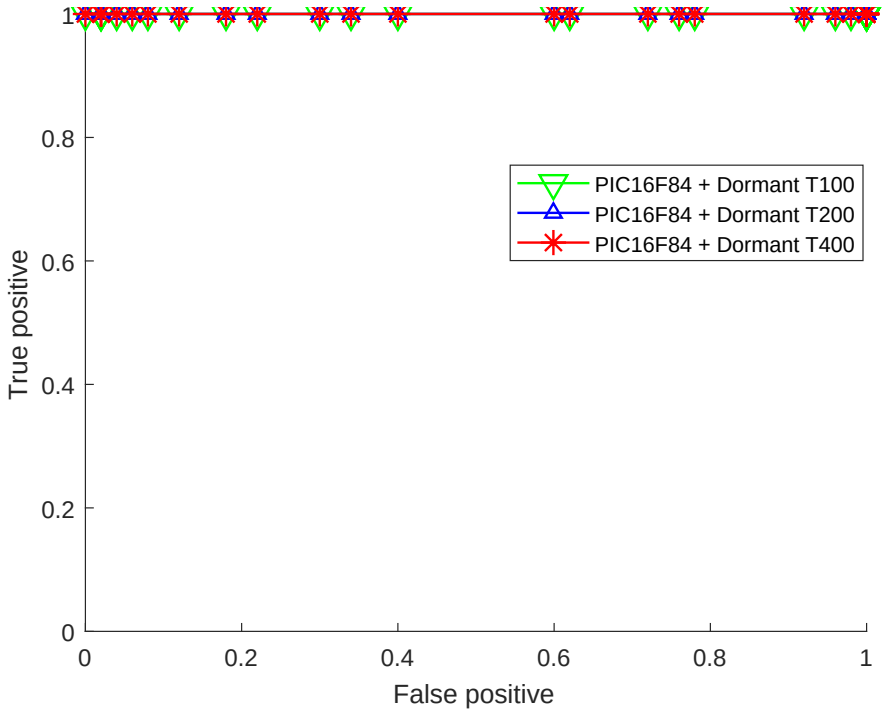


Figure 3.22: ROC curves for different Trojans on PIC16F84 circuit.

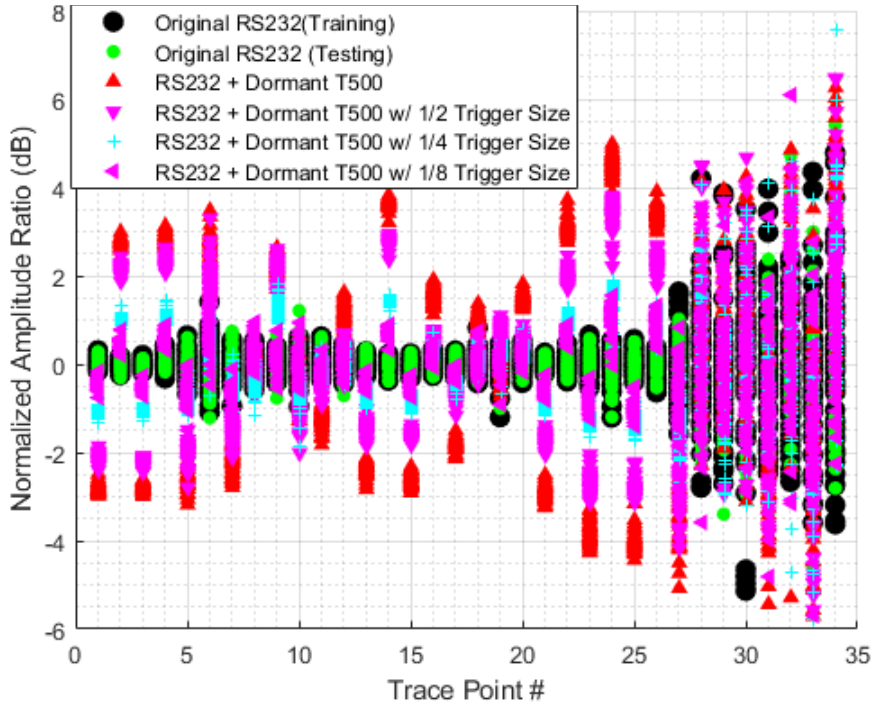


Figure 3.23: Normalized amplitude ratios for different trigger size of RS232 benchmarks.

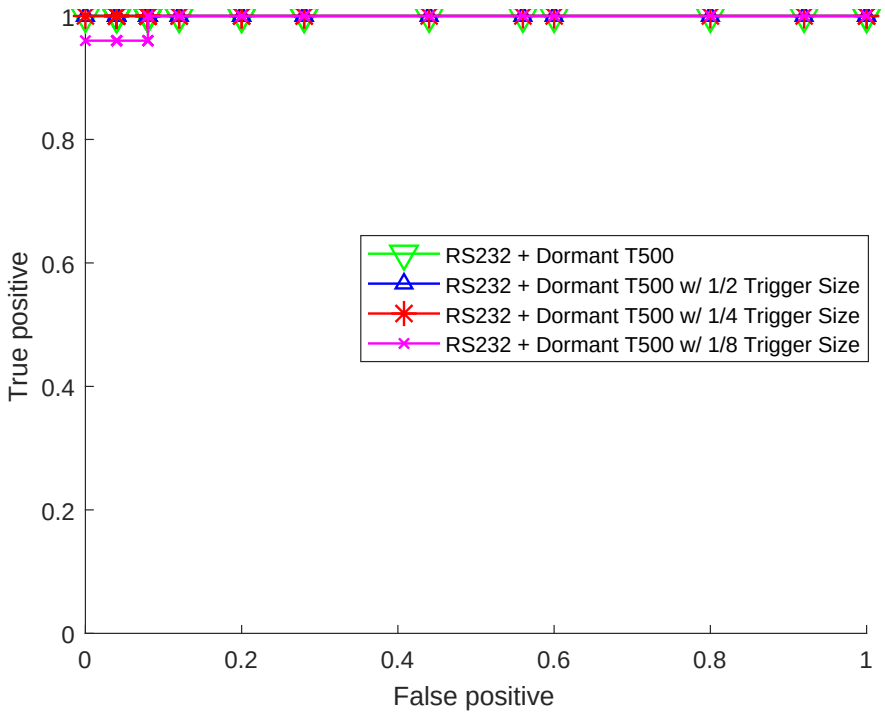


Figure 3.24: ROC curves for different trigger size of RS232 benchmarks.

CHAPTER 4

A COMPARISON OF BACKSCATTERING, EM, AND POWER SIDE-CHANNELS AND THEIR PERFORMANCE IN DETECTING SOFTWARE AND HARDWARE INTRUSIONS

4.1 Overview

As we discussed in Chapter 1, side-channel analysis is a powerful tool from both an attacker's and defender's perspective. Attackers use side-channels to circumvent traditional access controls and protections by exploiting the observable side effects of computation rather than attacking the computation's functionality. Defenders use side-channels for tracking program activities on various code levels such as loops, paths, basic blocks, and individual instructions, as well as for hardware Trojan detection. Understanding similarities and differences among types of side-channels is a necessary step in better utilization of side-channels. Therefore, to address this problem, we model and quantitatively compare the backscattering, EM, and power side-channels and their performance in detecting malware and HTs.

We start by describing the backscattering side-channel and comparing it with EM and power side-channels, two of the more widely used types of side-channels. Then, we characterize, model, and compare spectral characteristics of all three side-channels. Finally, we compare the performance of all three side-channels in detecting malware and HTs. The results show that for larger changes in the signals, such as those caused by malware intrusions, all three side-channels perform similarly. However, when smaller changes need to be observed, such as those caused by HTs, the backscattering side-channel outperforms the EM and power side-channels.

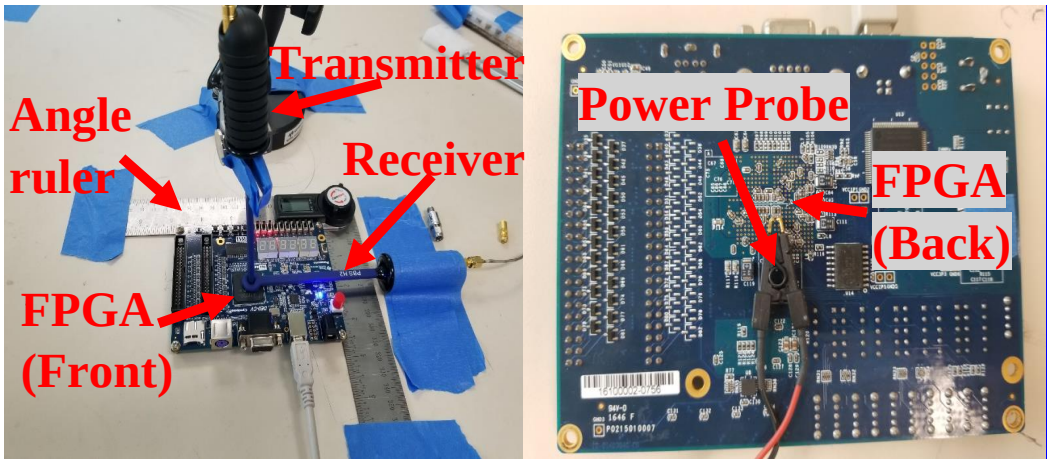
The rest of this chapter is organized as follows. Section 4.2 details the proposed signal

models for each side-channel, and verifies them against measurements. Section 4.3 compares how each side-channel's signal changes with different measurement parameters, such as distance and input power. Section 4.4 illustrates how malware and HT can be detected using each side-channel and compares their performance. Finally, Section 4.5 concludes the chapter.

4.2 Side-Channel Waveform Model Comparison

This section characterizes the three side-channels by proposing a waveform model and comparing it to the measurement results. The backscattering side-channel is a consequence of the impedance changes in digital circuits, which are caused by the transistors' two-state impedances reflecting a modulated signal. For each gate that switches, the impedance change persists for the rest of the cycle. As a result, the impedance switching causing the backscattering side-channel occurs in discrete steps, creating a signal similar to square pulse wave. However, the EM and power are consequences of the variation of the current flow in a circuit. As a gate switches, the current will be charged or discharged quickly, which means the current flow is a burst of current in brief duration. This burst of current does not persist for the rest of the cycle, and the duration is not the same from cycle to cycle. Therefore, this current change creates a signal whose amplitude and pulse width vary every cycle.

Starting with the backscattering side-channel, we change the duty cycle of the toggling circuit and measure the side-channel signal in frequency domain. As shown in Section 3.2 in Chapter 3, the switching activities of the toggling circuit create peaks at multiple harmonics of the toggling circuit's switching frequency in the measured side-channel signal. We then extract the first three harmonics and compare the shape of these harmonics with the shape of the first three harmonics of an ideal square pulse produced by simulation. Similarly, the shape of the first three harmonics when using EM and power side-channels are compared to those obtained through simulation. Finally, we compare the shape of the



(a) (b)

Figure 4.1: (a) The measurement setup for the backscattering and EM side-channels; (b) the measurement setup for the power side-channel.

first three harmonics obtained by backscattering, EM, and power measurement of the same circuits, respectively.

The measurement setup for the backscattering side-channel is shown in Fig. 4.1 (a). The setup includes a transmitter probe connected to a signal generator and a receiver probe connected to a vector signal analyzer. Both near-field probes are from the AARONIA Near Field Probe Set [103]. An Agilent MXG N5183A Signal Generator with an output power of 15 dBm is used as a signal source and an Agilent MXA N9020A Vector Signal Analyzer is used to record the signals. Note that both the signal generator and signal analyzer can be substituted by a single software defined radio (SDR) that has both transmitter and receiver. The devices under test (DUT) are Altera DE0 Cyclone V FPGA development boards. An angle ruler is used to position the FPGA boards and a laptop is used to control the devices and automate the measurements. The EM emanations are recorded by using the receiver probe in Fig. 4.1 (a). For the power measurements, the power signal is collected using an N7020 Power Rail Probe PBS2 [104] positioned on a capacitor in the chip, as shown in Fig. 4.1 (b).

As discussed in the Section 3.2 in Chapter 3, the toggling circuit toggles an array of

flip-flops between output-on (1) and output-off (0) states. When a flip-flop is turned on (1-state), its pull-up circuits are closed (conductive) and its pull-down circuits are open (high impedance). Conversely, when a flip-flop is turned off (0-state), its pull-up circuits are open and its pull-down circuits are closed. Because the impedances of the pull-up and pull-down circuits are not identical, the equivalent impedance of the circuit when flip-flops are in the 1-state is different from the impedance when flip-flops are in the 0-state. Therefore, the toggling circuit is switching between two different impedance levels at a particular frequency. The carrier signal is modulated by this difference in impedance levels, creating the sideband at the switching frequency from the carrier in the backscattered signal. Therefore, we can model the toggling-modulated backscattering signal as a sinusoidal carrier waveform modulated by a square-wave waveform that has a 0.5 duty cycle.

We generated an ideal square wave in MATLAB. By varying the duty cycle of this square pulse train and recording the magnitudes of the first three harmonics, we obtain the function of how the harmonics change with the duty cycle, as shown in Fig. 4.2. Note that when the duty cycle is 0.5, the amplitude of the second harmonic drops very low; therefore, it is not shown in the figure.

We compare this model with the measurements produced by the toggling circuit with the same toggling frequency that was used in the square-wave model. Furthermore, we change the duty cycle and record the power of different harmonics to verify that their measured shapes match the model's predictions when the duty cycle of the modulating square-wave signal is changed. Fig. 4.3 shows how each of the first, second, and third harmonics change with the duty cycle. These results are very similar to those predicted by the ideal-square-wave model in Fig. 4.2.

Next, we provide signal model for the EM and power side-channels and then compare the model's predictions with measurements. In the analysis, we use the same toggling circuit that was used for the backscattering side-channel.

Here, the current magnitude and pulse width depend on the number of flip-flops that

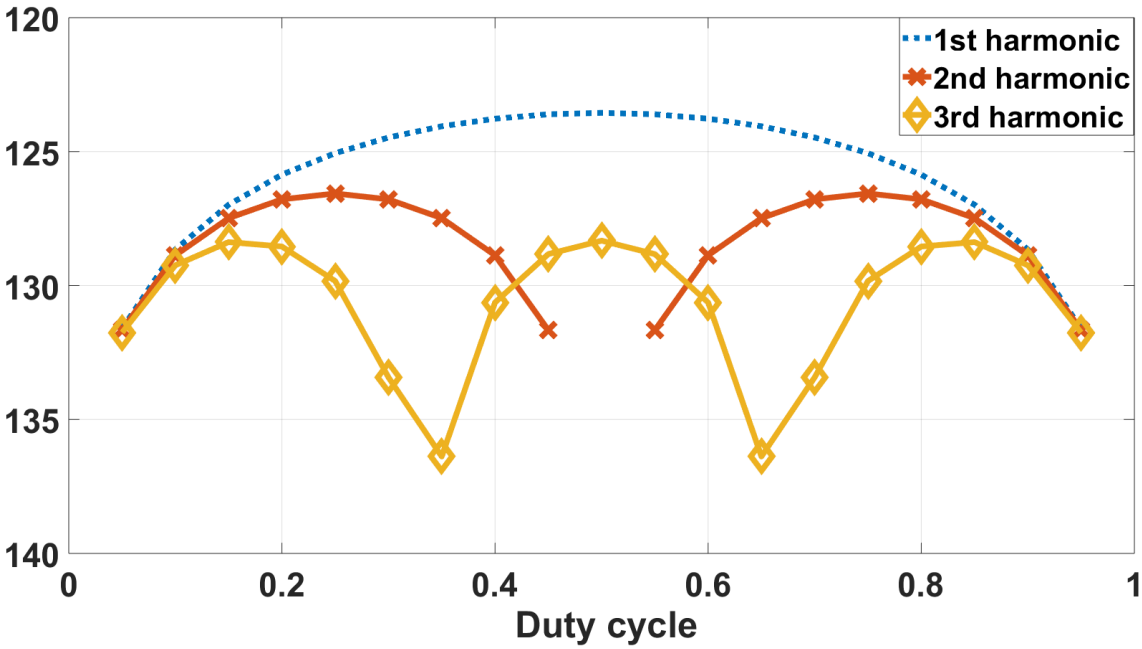


Figure 4.2: Changes in the first three harmonics of a modeled ideal square pulse as a function of duty cycle.

are active during each clock cycle. Since the current's amplitude and pulse width vary every cycle, the EM and power side-channels signals' amplitude and pulse width change every cycle. Therefore, the side-channel signal can be modeled as a waveform that has a varying amplitude and pulse width, as shown in Fig. 4.4. The first three harmonics of this pulse train and how they change with the duty cycle are shown in Fig. 4.5. To verify this model, we again use the toggling circuit switching at f_{m} = 1.25 MHz. We vary the duty cycle of the toggling circuit and record the first three harmonics of the EM and power side-channel signals. The results are shown in Figs. 4.6 and 4.7, respectively. The results demonstrate that the EM and power side-channels have very similar harmonic amplitude trends and follow the behavior of the model in Fig. 4.5. On the other hand, these trends are different from the backscattering side-channel's results in Fig. 4.3, demonstrating that the backscattering side-channel is different from the EM and power side-channels.

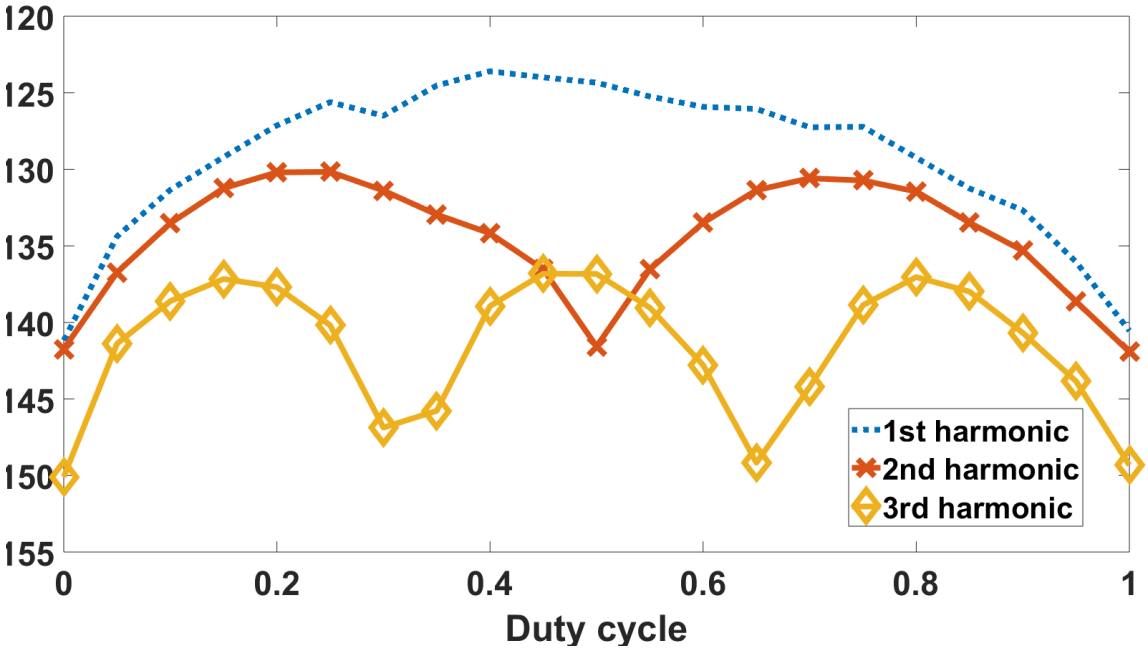


Figure 4.3: The first three harmonics of the measured backscattered signal as a function of duty cycle.

4.3 Comparison of the Characteristics of the Backscattering, EM, and Power Side-Channels

4.3.1 Impact of Distance on the Side-Channels

While the power side-channel can not be measured from a distance, both the backscattered and EM signals can be recorded from several meters away. The longer the distance, the weaker the signal is in both side-channels. However, the backscattering side-channel signal's strength can be increased by increasing the power of the carrier wave. This allows for measurements over a longer distance.

In this section, we analyze how the magnitudes of the backscattering and EM side-channels' signals depend on the distance from the monitored device. Because it requires measuring from a distance, the near-field probes in Fig. 4.1 are replaced by horn antennas with an average gain of 9.1 dBi [105], as illustrated in Fig. 4.8. The toggling circuit with a toggling frequency f_{m} = 900 kHz is used and the distance from the antennas to the DUT is swept from 50 cm to 3 m. The carrier frequency is 10.91 GHz because it does not in-

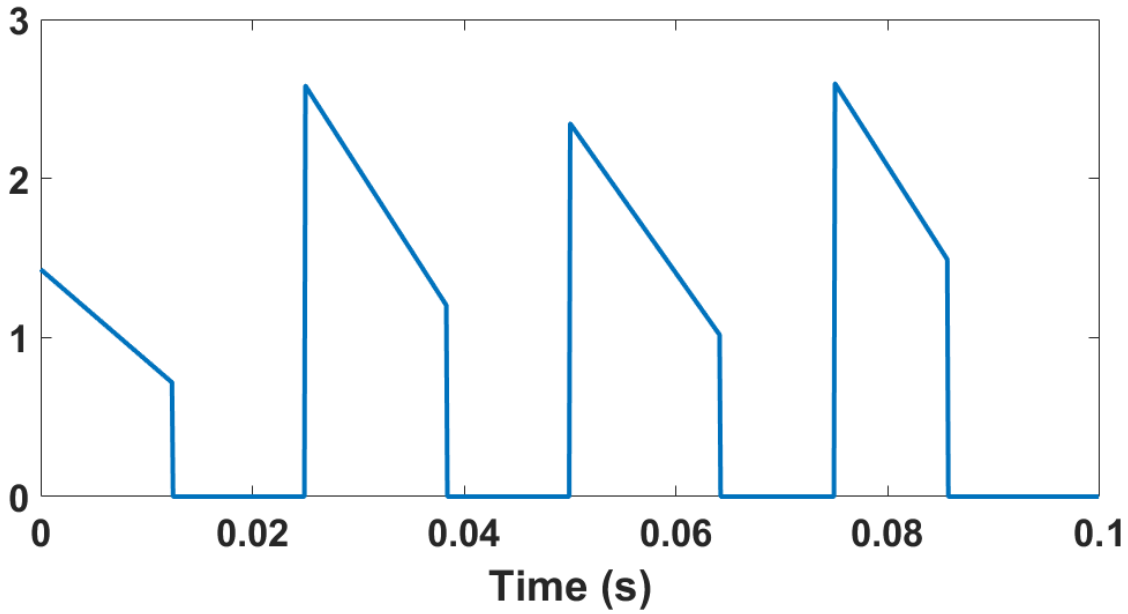


Figure 4.4: Waveform model of the EM and power side-channels signal.

terfere with other periodic signals on the DUT, and the horn antennas give high gain at the frequency. We measure the power of the first harmonic sideband caused by the toggling circuit. The power of the carrier is kept at 15 dBm, the maximum power that the signal generator can generate. The results in Fig. 4.9 demonstrate how the magnitudes of the backscattering and EM side-channel power (in dB) change with the distance. Unsurprisingly, signal strength decays rapidly with the distance in both cases.

4.3.2 Impact of Carrier Input Power on the Side-Channels

For the EM and power side-channels, the strength of measured signal is limited by the strength of the signal leaked from the device. In contrast, the backscattering side-channel does not have this limitation since it does not depend on leakage from the circuits. In this section, the impact of carrier input power on the quality of the backscattering side-channel is analyzed. In this measurement, the toggling circuit has f_{m} = 1.25 MHz, and the power of the carrier is swept from -15 dBm to 3 dBm. The results in Fig. 4.10 show that the sideband power increases as the transmitted carrier power increases. This means that we

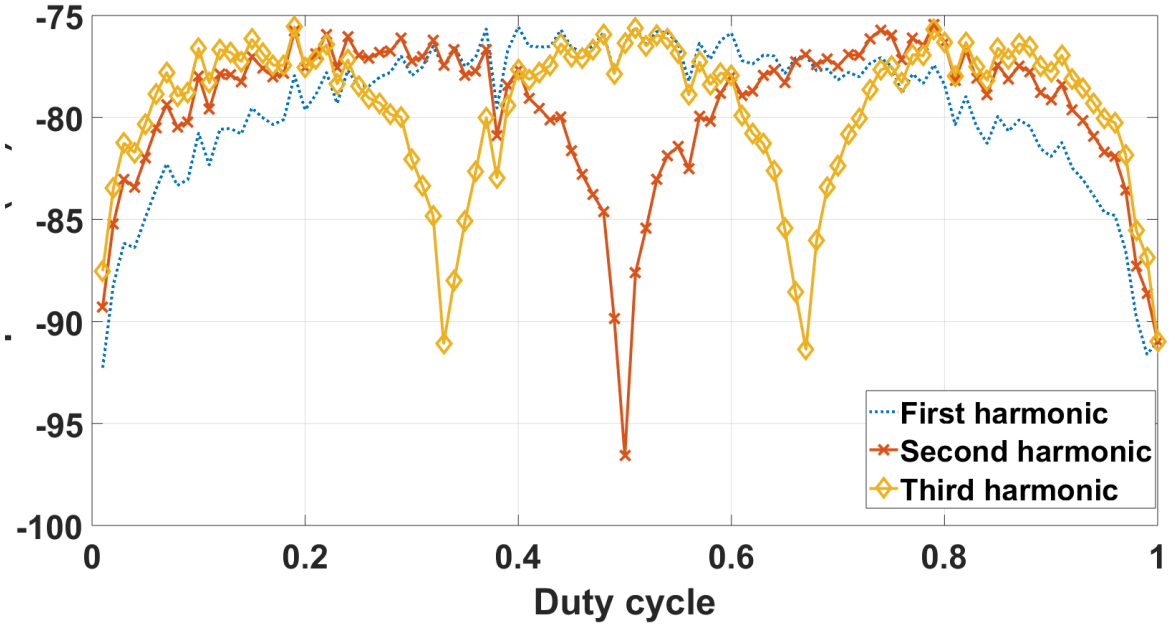


Figure 4.5: The change in the first three harmonics of a modeled current pulse as a function of duty cycle.

can control the strength of the backscattered signal by changing the power at the carrier's transmitter. For example, when measuring far away from the device, or when there are obstacles between the antenna and the device, we can compensate for the path loss by transmitting a stronger carrier. Therefore, unlike the EM and power side-channels, the backscattering side-channel is an "active" side-channel.

4.4 Comparison of the Backscattering, EM, and Power Side-Channels in Detecting Software and Hardware Intrusions

4.4.1 Comparison of Backscattering-Based, EM-Based, and Power-Based Software Malware Detection

In this section, we use the EM-Based Detection of Deviations in Program Execution (EDDIE) technique, a malware detection prototype from [53], to demonstrate how these side-channels can be used to detect malware, and then compare their performance. EDDIE is a novel method for monitoring software activity without paying any overhead on hardware,

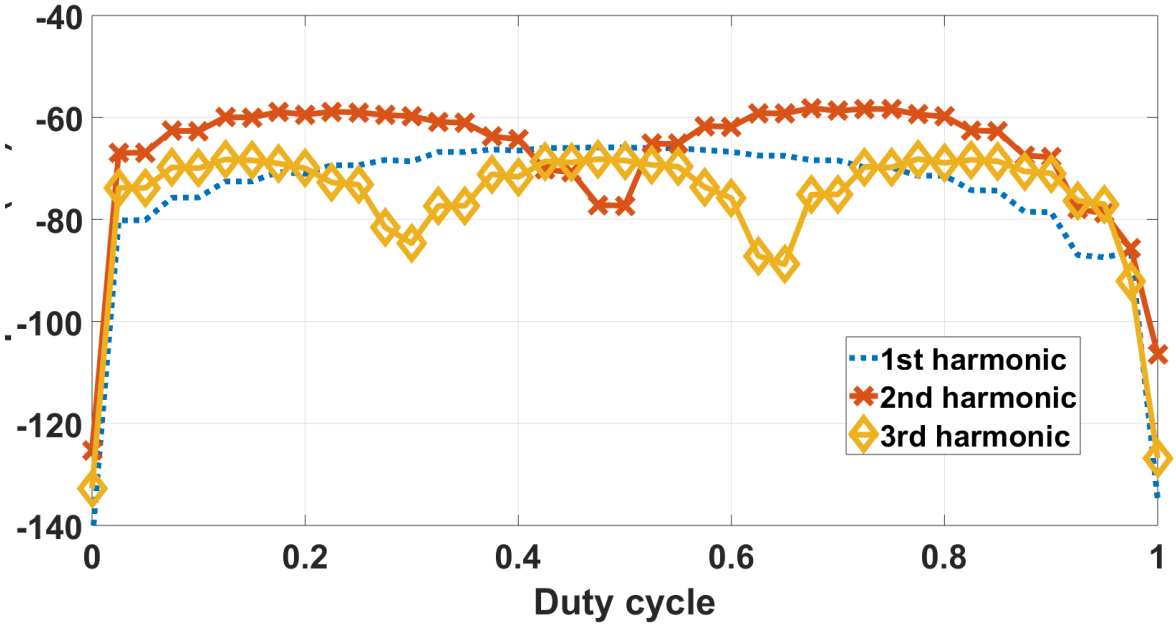


Figure 4.6: The first three harmonics of the measured EM side-channel signal as a function of duty cycle.

software, and resources on the monitored system [53]. In [53], the authors only exploit the EM side-channel for EDDIE without considering any other side-channel. In this paper, we use the EDDIE method for all three side-channels and make a comparison. Note that this study is intended only as a comparison, and that a much more detailed study of EDDIE's EM-based performance can be found in [53].

EDDIE relies on spikes, or peaks, in the frequency spectrum generated by periodic activity, such as a loop in a software program, to monitor and detect malicious behavior. As discussed in [53], loops tend to produce peaks in the spectrum that correspond to their per-iteration timing. These peaks are strongest when an existing periodic signal, such as a clock signal, is amplitude (AM) modulated by changes in processor activity. For example, Fig. 4.11 illustrates a spectrum of the AM modulated loop activity of a software program. In the figure, the center peak, the strongest peak, is the clock signal. The loop's per-iteration execution time is approximately $T \approx 20.8\ \mu$s. The spectrum shows that there are peaks at the frequency of $f = 1/T \approx 48$ kHz and its harmonics from the center clock signal. These

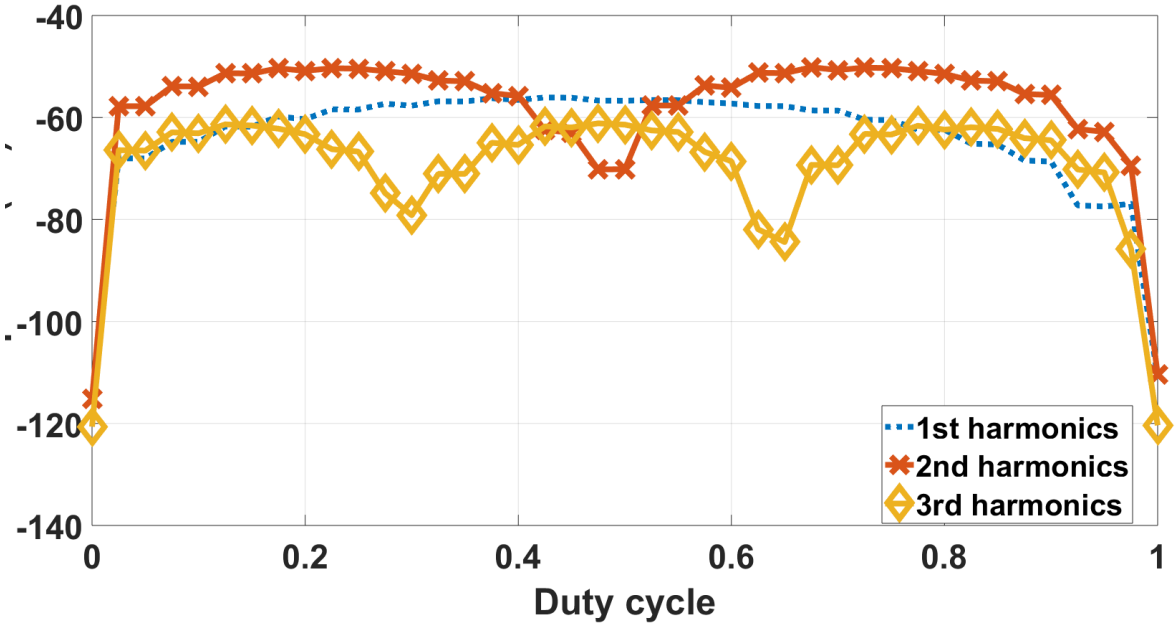


Figure 4.7: The first three harmonics of the measured power side-channel signal as a function of duty cycle.

peaks appear when the loop starts and disappear when the loop ends.

EDDIE takes the time-domain signal and converts this signal into a sequence of spectra using the Short-Term Fourier Transform (STFT), as shown in Fig. 4.19. Fig. 4.19 shows a spectrogram of a program consisting of seven loops. A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. The x-axis is the frequency domain, and the y-axis is the time domain. In the figure, the seven sets of peaks correspond to the seven loops in the program. They appear in chronological order (loop 1 to loop 7) from the top to the bottom of the spectrogram.

EDDIE includes two phases: training and testing, as illustrated in Fig. 4.12. In the training phase, EDDIE first analyzes the program to identify the loops, inter-loop regions, and their orders. This is done in LLVM (Low-Level Virtual Machine) compiler by adding a pass which builds the loop-level control flow graph. Then, it collects a set of side-channel traces while the program runs multiple times to create a "golden" set of traces. In this situation, the program is a clean copy that is verified to be malware-free. Next, it converts

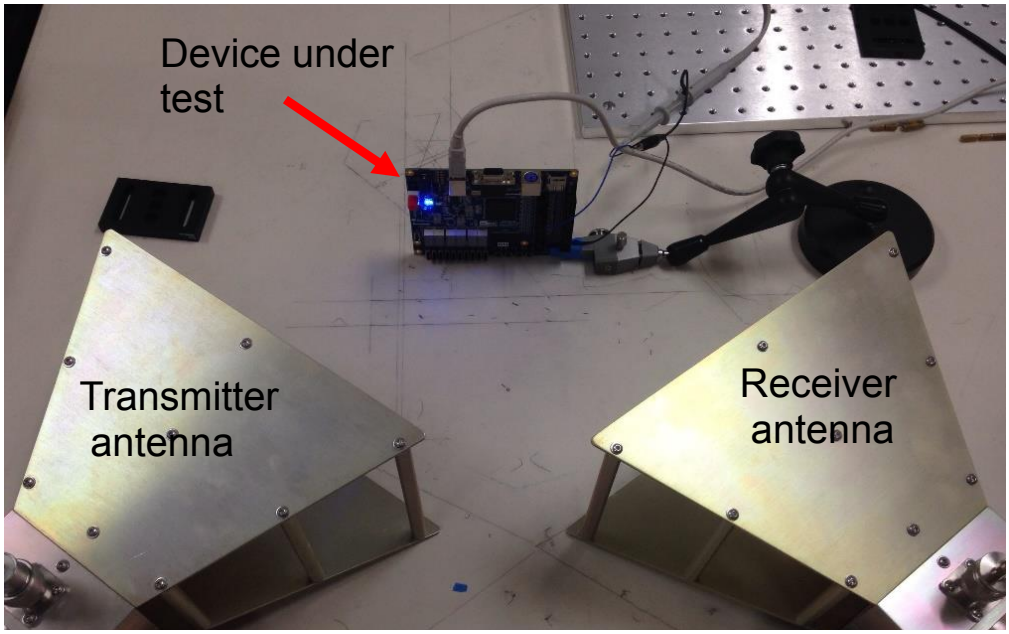


Figure 4.8: Setup for the backscattering and EM measurements from a distance.

each trace into a series of sample spectra, in which each spectral sample corresponds to a 1 ms window of time with 75% overlap with the previous sample. Then, it categorizes each spectral sample as belonging to a particular part of the program. For example, a run for a program comprised of 7 loops executed will have 9 spectral sample categories: One for the execution before the first loop, seven for the seven loops, six for inter-loop transitions, and one for execution after the last loop. After categorizing, EDDIE extracts the peak features for each spectral sample in each category.

In the testing phase, EDDIE collects the side-channel signal traces while the program is running, converts the signal into a series of sample spectra, extracts the peak features from these spectral samples, and then uses the Kolmogorov-Smirnov (K-S) test to decide, at a certain level of confidence, how unlikely the testing samples belong to the same distribution as the training samples. If there is high enough confidence, EDDIE flags it as malware-infected. An important parameter in the EDDIE's K-S test is the number of time-consecutive samples that will be tested in each test. If the number is small, the K-S test only needs a small set of time-consecutive samples to give a decision, hence the detection

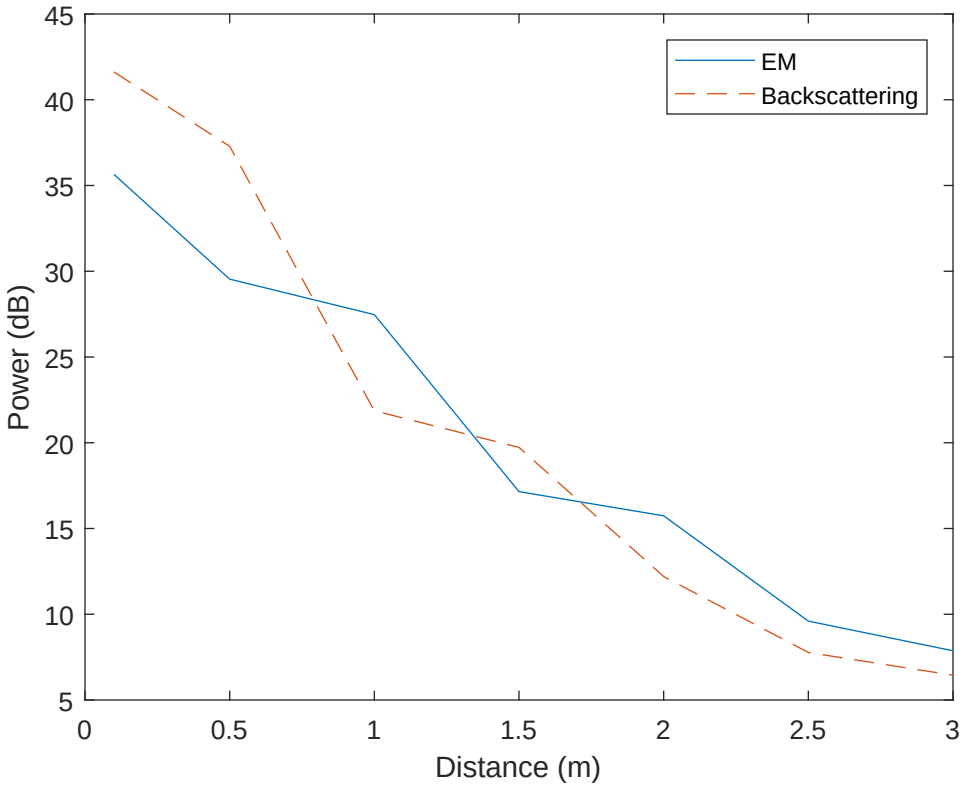


Figure 4.9: Side-channel power as a function of the distance from the DUT.

latency is low but the detection accuracy is also low. On the other hand, when the number increases, the detection accuracy increases but the latency also increases. For a given application and side-channel, the number of consecutive samples (n_s) needed for K-S test can be found during training by incrementally increasing it until getting zero false positive.

EDDIE's algorithm starts with taking the first n_s spectral samples of the tested signal and testing it against the training samples and continues to do so (by adding a new sample and removing the oldest one) until the test does not pass (malicious activity reported) or all spectral samples of the tested signal have been used.

Measurement Setup

Malware can be injected to attack a software anytime the software is executed. Because it is not always possible to get close to the device when it is running, it is highly desirable for a malware detection technique to be able to monitor and detect malware from a distance.

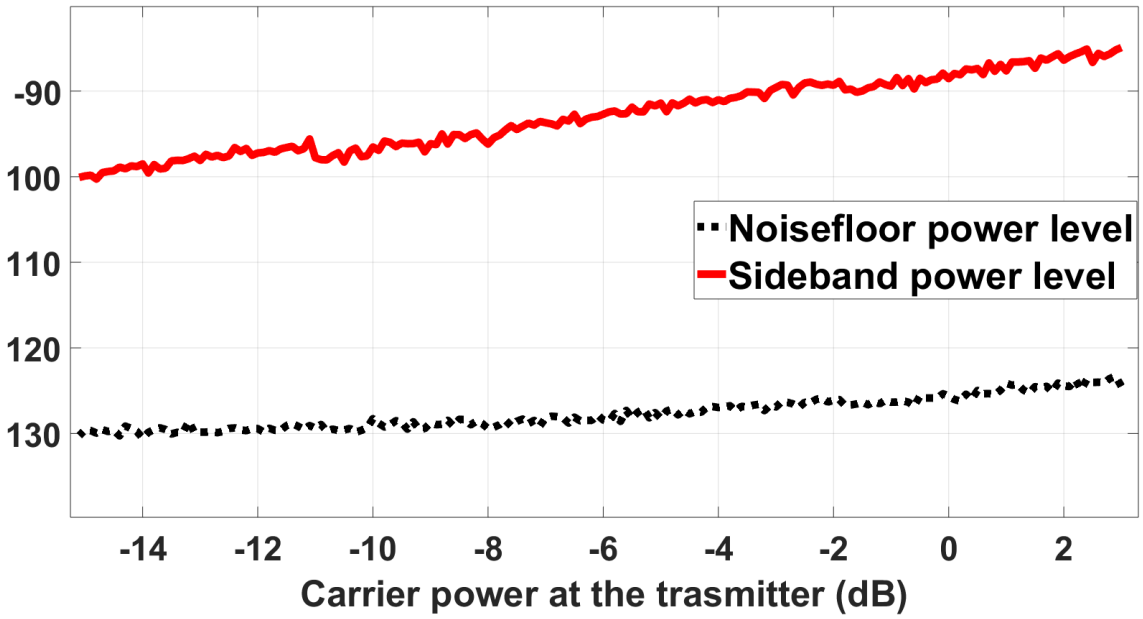


Figure 4.10: The received power of the backscattered signal as a function of the carrier power at the transmitter.

Therefore, we use horn antennas [105] to demonstrate the ability to perform EDDIE from a distance for both the backscattering and EM side-channels. For the power side-channel, it is not possible to extract information from a distance; therefore, we use a power probe placed on the device for the measurements.

The setup for the backscattering side-channel is shown in Fig. 4.8. The setup includes a transmitter horn antenna connected to a signal generator (for the carrier signal), and a receiver horn antenna connected to a spectrum analyzer that analyzes the received signal. Both antennas are placed 15 cm from the DUT. For the EM side-channel, the EM emanations are recorded using the receiver horn antenna in Fig. 4.8. For the power side-channel, the power signal is collected by using the setup shown in Fig. 4.1 (b), which is described in Section 4.2. The DUT is, again, an Altera DE0 Cyclone V FPGA development board. For each experiment, the FPGA implements a Nios-II soft processor.

Compared to the EM and power side-channels, the backscattering side-channel needs a transmitter. However, by using SDRs, except for an extra antenna, the setup overhead for

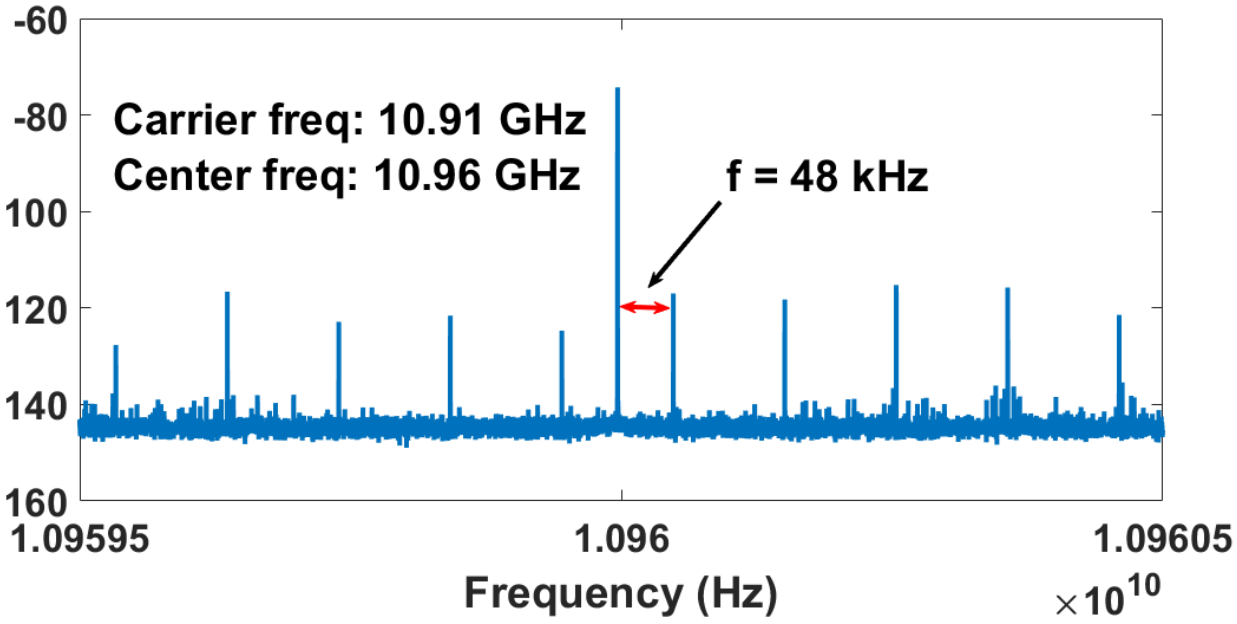


Figure 4.11: The spectrum of a loop structure in a software program.

the backscattering side-channel is the same as for EM side-channel.

Regarding stealthiness, compared to the EM and backscattering side-channel, it is more difficult for the power side-channel to be measured stealthily because power side-channel signal cannot be obtained from distance. Both EM and backscattering side-channel measurement can be performed from a distance, but it might be harder for attackers to monitor the backscattering side-channels stealthily because it needs to transmit a signal toward the device. However, for defense and intrusion detection purpose, which are the main focus of this paper, stealthiness is not a concern.

To create test benchmarks, we implemented the bitcount program from MiBench [106], which consists of seven loops, on the FPGA. The runtimes for loop 1 to 7 are 3,568 ms, 2,632 ms, 3,146 ms, 1,390 ms, 2,250 ms, 2,900 ms, and 2,480 ms, respectively. We use three different types of malware, which are proof-of-concept implementations of Ransomware, DDOS, and Stuxnet-like malware, respectively. Stuxnet-like malware is a class of malwares that have similar behavior and attack mechanism to the Stuxnet malware. In a Stuxnet-like attack, the adversary modifies the code to change a critical value based on

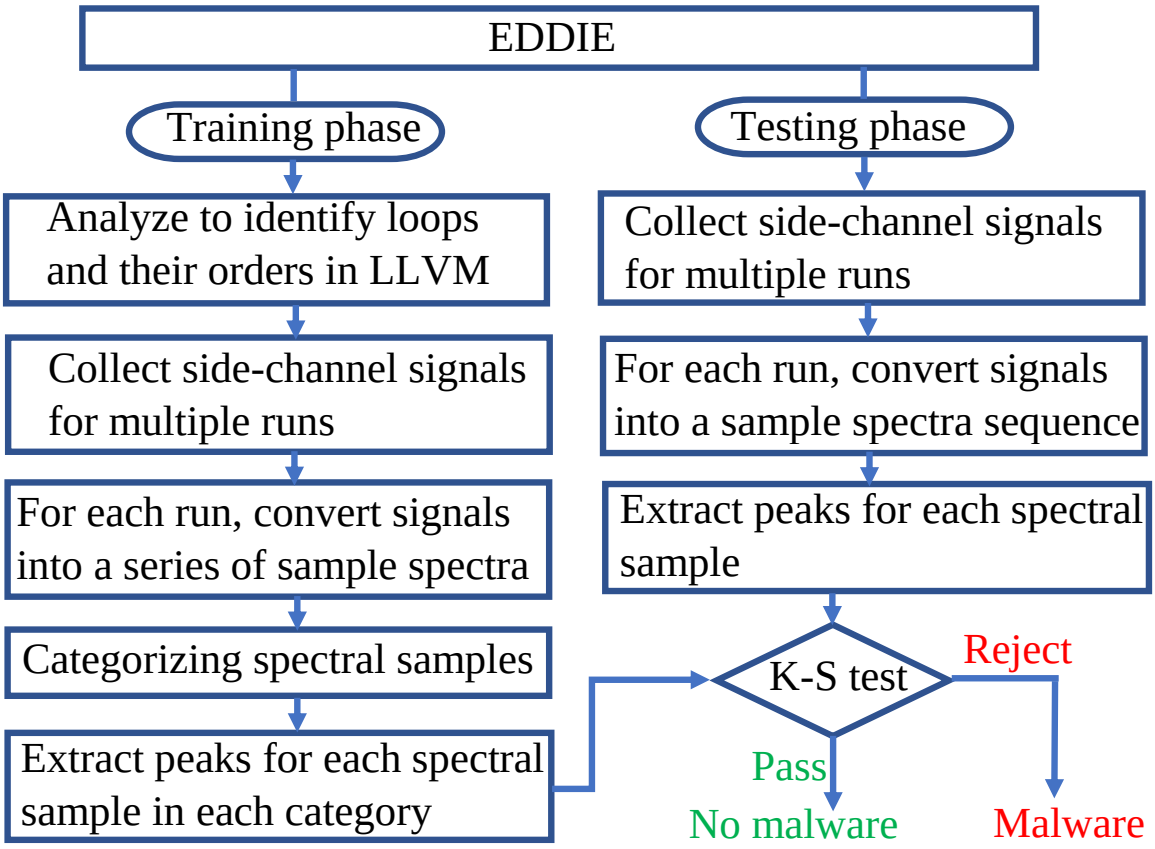


Figure 4.12: Overview of EDDIE.

some conditions. In order to mimic the behavior of Stuxnet-like malwares, we add a small piece of code to the source code of the program to form a Stuxnet-like attack. The malwares are injected to the bitcount program between the loops 2 and 3. We also do an experiment that we inject the Stuxnet-like malware inside the loop 3 of the bitcount program. Intuitively, if a malware is injected inside a loop, it will be much easier to detect than ones outside loops because the malware will affect all the spectral samples created by the loop. Therefore, when injecting the Stuxnet-like malware inside the loop 3, we reduce its size as shown in Table 4.1 to make it more meaningful to detect.

Fig. 4.13 shows the spectrogram of loop 2, loop 3, and a loop-to-loop transition region between loop 2 and loop 3 of a normal (malware-free) bitcount program. As illustrated in Fig. 4.13, a loop-to-loop transition consists of non-repetitive (non-loop) code that normally

executes briefly compared to repetitive (loop) regions. This is because modern processors, even inexpensive ones, are capable of executing hundreds of millions of instructions per second. As a result, a non-repetitive execution that covers an entire 1 ms window requires several megabytes of executable code. Typically, a loop-to-loop transition typically lasts less than one 1 ms spectral sample.

Figs. 4.14-4.16 show spectrograms of loop 2, loop 3, and a loop-to-loop transition region between loop 2 and loop 3 of a DDOS-infected, Ransomware-infected, and outside-loop Stuxnet-like malware-infected bitcount program, respectively. The horizontal dashed lines indicate where the attacks start and end in the spectrogram. The DDOS attack added 975 ms of execution to the program, introducing several additional spectral samples to the loop-to-loop transition region in the spectrogram. The difference caused by malware between the malware-infected spectrogram in Fig. 4.14 and normal spectrogram in Fig. 4.13 is easily identified by human eyes. The execution of Ransomware and outside-loop Stuxnet-like malware added 3,680 ms, and 92 ms to the execution of the program, respectively.

Fig. 4.17 shows spectrogram of loop 2, loop 3, and a loop-to-loop transition region between loop 2 and loop 3 of a bitcount program infected by inside-loop Stuxnet-like malware inside loop 3. The runtime of the inside-loop Stuxnet-like malware is only 0.0039 ms. However, since it is injected inside loop 3, it affects all the spectral samples created by loop 3. As shown in Fig. 4.17, the malware shifted the peak of loop 3 to a different frequency in the spectrogram. Table 4.1 summarizes the runtime of all the malwares.

Table 4.1: Summary of the impact of malwares

	Runtime (ms)
DDOS	975
Ransomware	3,680
Outside-loop Stuxnet-like malware	92
Inside-loop Stuxnet-like malware	0.0039

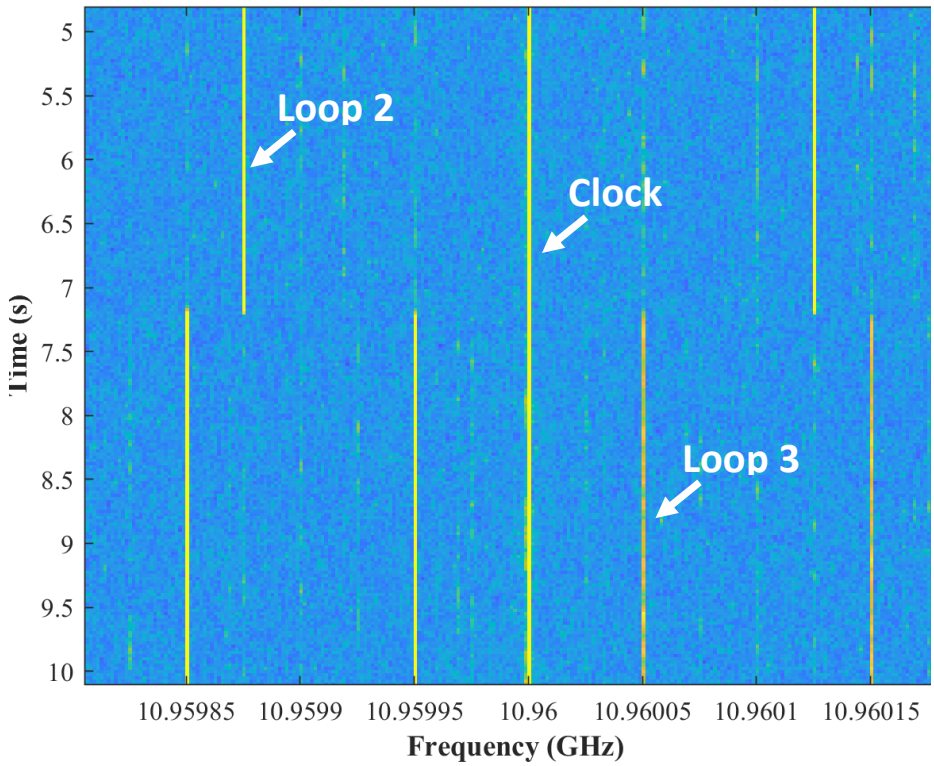


Figure 4.13: The spectrogram of malware-free bitcount software zoomed in loop 2 and 3.

Evaluation

For each of the three side-channels, we run the genuine bitcount program and record 25 traces of the respective side-channel signal for training. For testing, we first measure 25 traces of the respective side-channel signal when running the genuine bitcount program, 25 traces of the respective side-channel signal when the bitcount program is infected by DDOS, 25 traces of the respective side-channel signal when the bitcount program is infected by Ransomware, 25 traces of the respective side-channel signal when the bitcount program is infected by outside-loop Stuxnet-like malware, and 25 traces of the respective side-channel signal when the bitcount program is infected by inside-loop Stuxnet-like malware.

The spectrograms in Figs. 4.18-4.20 show that all seven loops in the bitcount program can be observed (the most visible peak of each loop is highlighted in white) using the backscattering, EM, and power side-channels, respectively. Note that the strongest har-

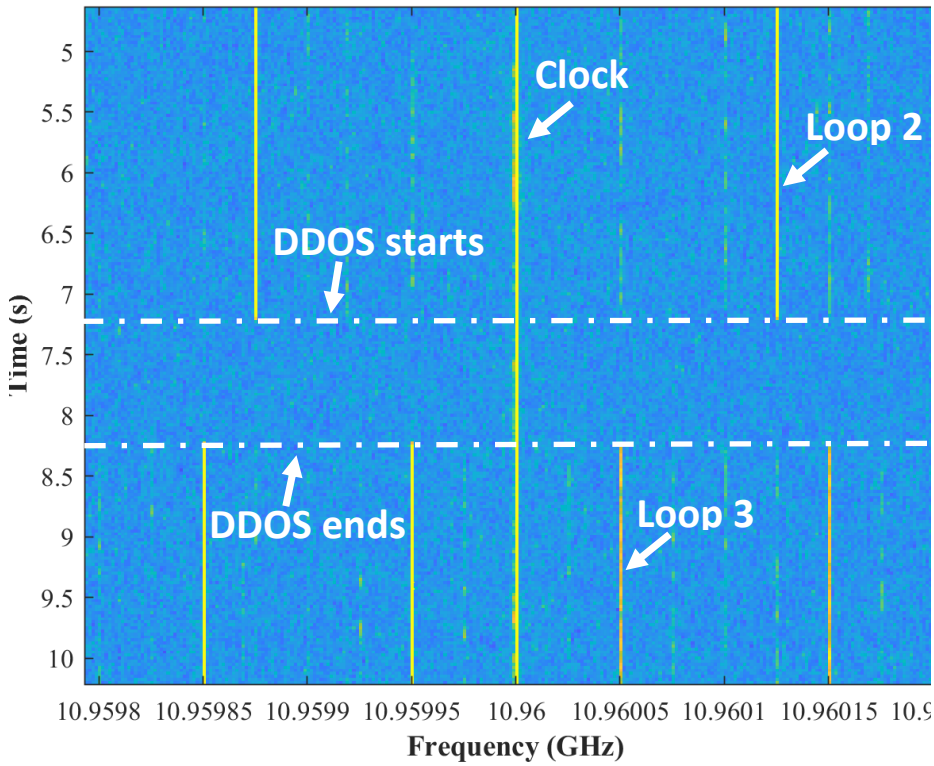


Figure 4.14: The spectrogram of the bitcount program infected by DDOS between loop 2 and 3.

monics for each loop could be different for each side-channel.

EDDIE achieves 100 percent accuracy with zero false positive rates for all three side-channels and the four malware attacks. This result matches what was reported in [53]. The detection latency for the bitcount benchmark is 42 ms, which is the same for the backscattering, EM, and power side-channels. Because the variation in the processor activity caused by the malware is relatively large, the required sampling rate of the side-channels is not very high. As a result, even though the three side-channels have different sized bandwidths, they still work well for detecting the malwares. The power side-channel gives the strongest signal because it was measured by a probe that was contacted directly with the device, while both the backscattering and EM side-channels were measured by antennas from a distance. However, as mentioned earlier, the power side-channel does not support monitoring at a distance, while the backscattering and EM side-channels can be recorded from

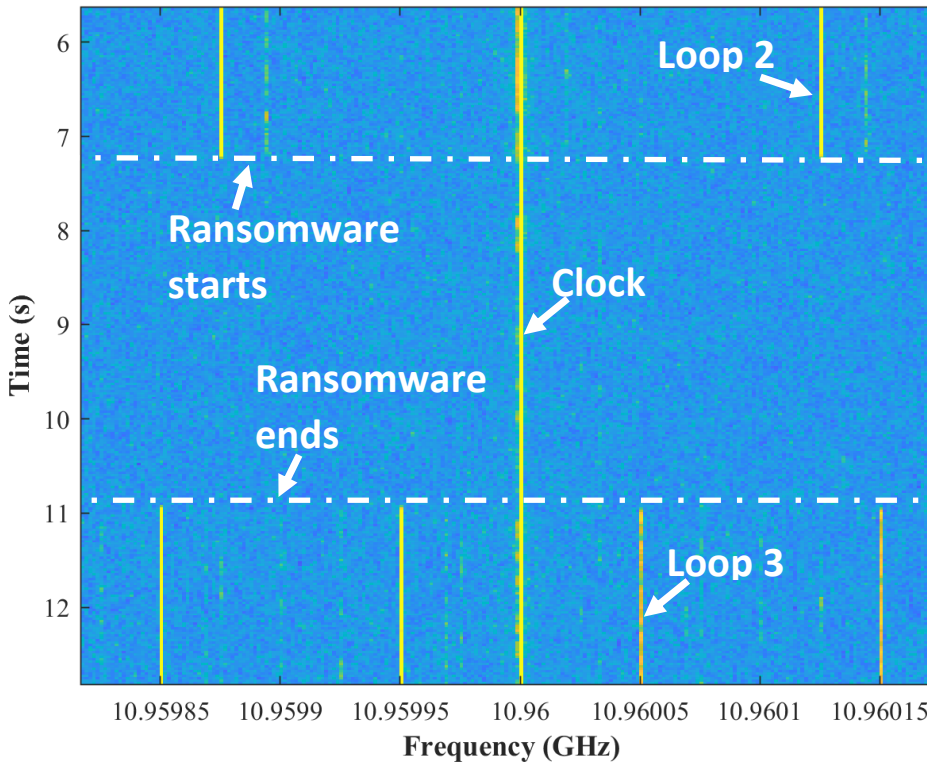


Figure 4.15: The spectrogram of the bitcount program infected by Ransomware between loop 2 and 3.

meters away. Furthermore, based on the spectrograms, the backscattering side-channel signals' loop were weaker than the signal's measured by the the EM and power side-channels. However, unlike the other two side-channels, the backscattering signals' strength can potentially be improved by increasing the transmitted carrier power.

To further evaluate and compare the three side-channels in the context of malware detection, we implement another program based on basicmath benchmark from MiBench [106], which consists of four loops, on the FPGA. The runtimes for loop 1 to 4 are 1,505 ms, 2,480 ms, 2,143 ms, and 2,592 ms, respectively. We use the same implementation of DDOS, Ransomware, outside-loop Stuxnet-like malware, inside-loop Stuxnet-like malware. We inject the first three malwares into the basicmath program between the loops 3 and 4, and the last one inside loop 4 of the basicmath program.

The spectrograms in Figs. 4.21-4.23 show that all four loops in the basicmath program

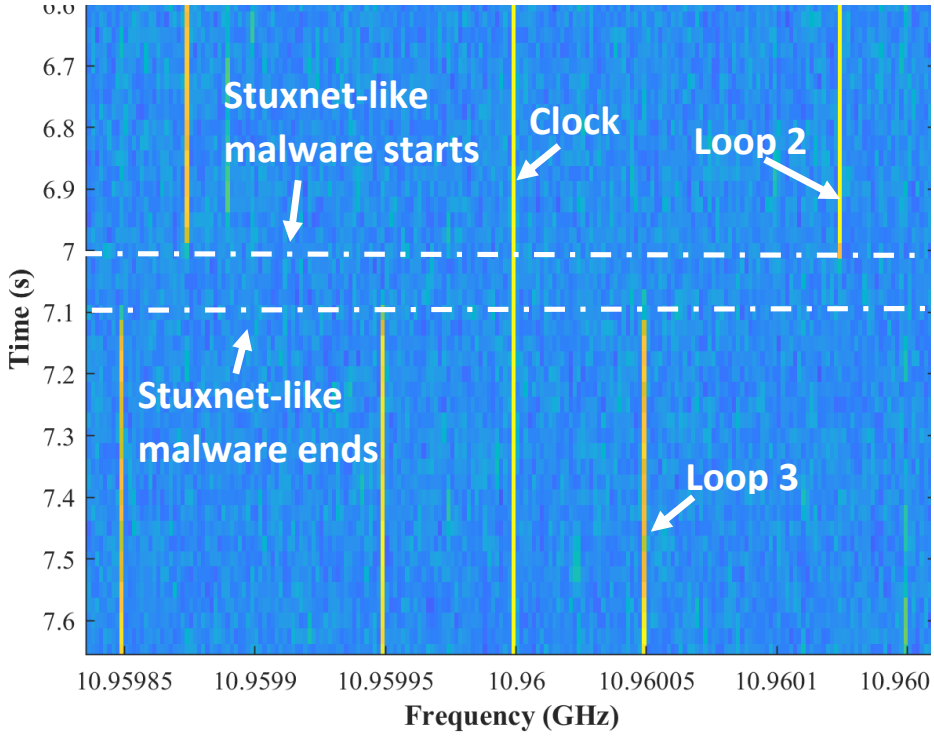


Figure 4.16: The spectrogram of the bitcount program infected by outside-loop Stuxnet-like malware between loop 2 and loop 3.

can be observed (the most visible peak of each loop is highlighted in white) using the backscattering, EM, and power side-channels, respectively. EDDIE achieves 100 percent accuracy with zero false positive rates for all three side-channels and the four malware attacks. The detection latency for the basicmath benchmark is 25 ms, which is the same for the backscattering, EM, and power side-channels.

4.4.2 Comparison of Backscattering-Based, EM-Based, and Power-Based Hardware Trojan Detection

Hardware Trojan Detection Method

In this section, we use the method proposed in [64] to compare performance of backscattering-based, EM-based, and power-based HT detection. This method relies on measuring the amplitude of the side-channel signal at the sidebands for the first m harmonics of the clock frequency. It then calculates amplitude ratios, i.e., the amplitude of a harmonic divided by

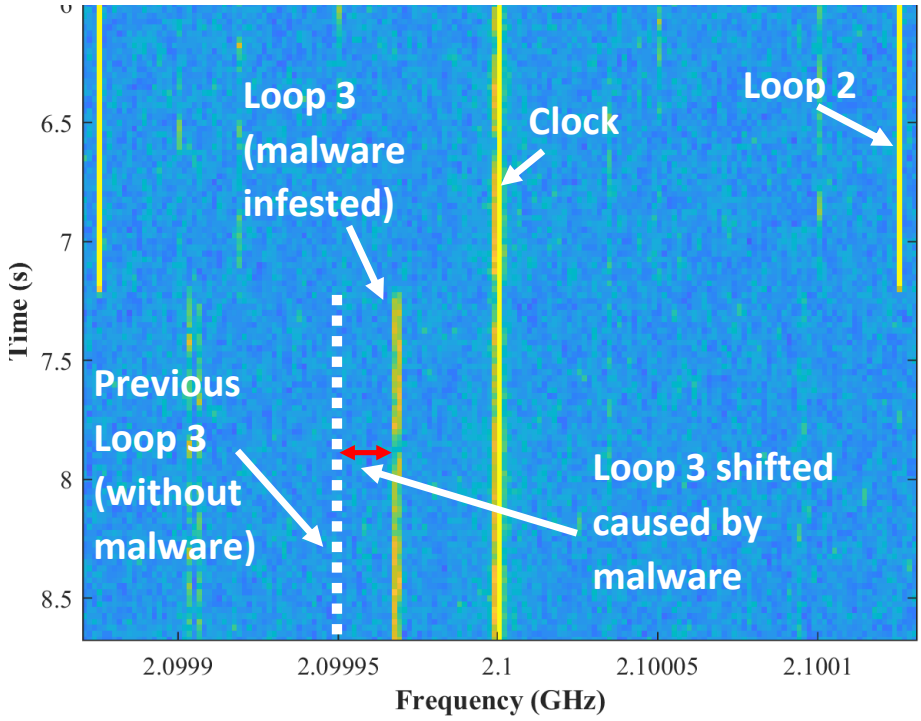


Figure 4.17: The spectrogram of the bitcount program infected by inside-loop Stuxnet-like malware inside loop 3.

the amplitude of the previous harmonic to cancel out the distance dependent attenuation factor. Note that each clock harmonic produces two sideband components. In [64], the authors measure points to the right of the carrier, i.e. $f_{\text{carrier}} + f_{\text{c}}$, $f_{\text{carrier}} + 2 * f_{\text{c}}$, etc. These m amplitudes are measured for a given circuit form a trace, and each trace characterizes the amount, duration, and timing of the circuit's impedance-changing activity during a clock cycle. The calculated $m - 1$ amplitude ratios are then used for comparing traces. By collecting training traces using one or more genuine ICs, it can detect HTs on other ICs by collecting their traces and comparing them with the training traces. The method consists of two phases: training and detection.

- Training: The training phase includes measuring the side-channel signal from a genuine IC K times. During each measurement, the amplitudes of the first m harmonics of the IC's clock are recorded. The $m - 1$ amplitude ratios are computed from these amplitudes. Then, for each ratio, the mean and standard deviation across the K measurements are computed, and the detection threshold for HT detection is computed

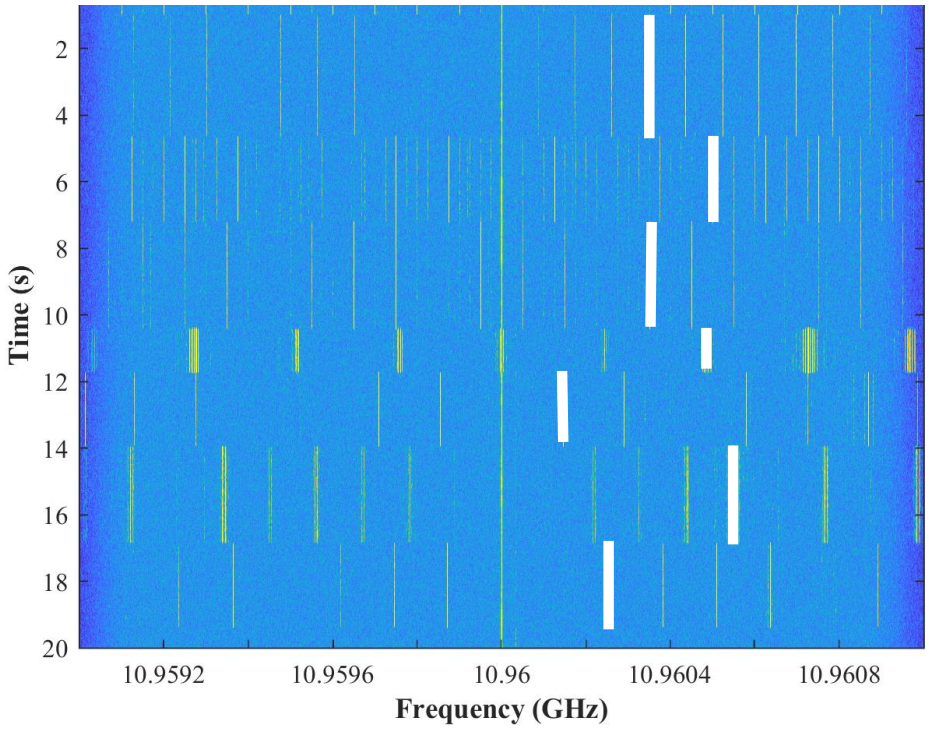


Figure 4.18: The spectrogram of bitcount software received via the backscattering side-channel.

as the sum of the $m-1$ standard deviations.

- Detection: First, m amplitudes are recorded and $m-1$ amplitude ratios are computed from these amplitudes for the IC under test. Then, for each of the $m-1$ amplitude ratios, we calculate the absolute difference between it and the corresponding mean computed during training. Finally, we sum the difference of all $m-1$ amplitude ratios and the sum of these deviations is compared to the sum of standard deviations from training. The IC under test is labeled as HT-free if its sum of amplitude-ratio deviations is lower than this detection threshold (the sum of standard deviations from training).

Measurement Setup

Because HTs cannot be injected after the chip is fabricated, all the chips need to be tested before selling to customers. It is not necessary to perform HT tests from a distance because

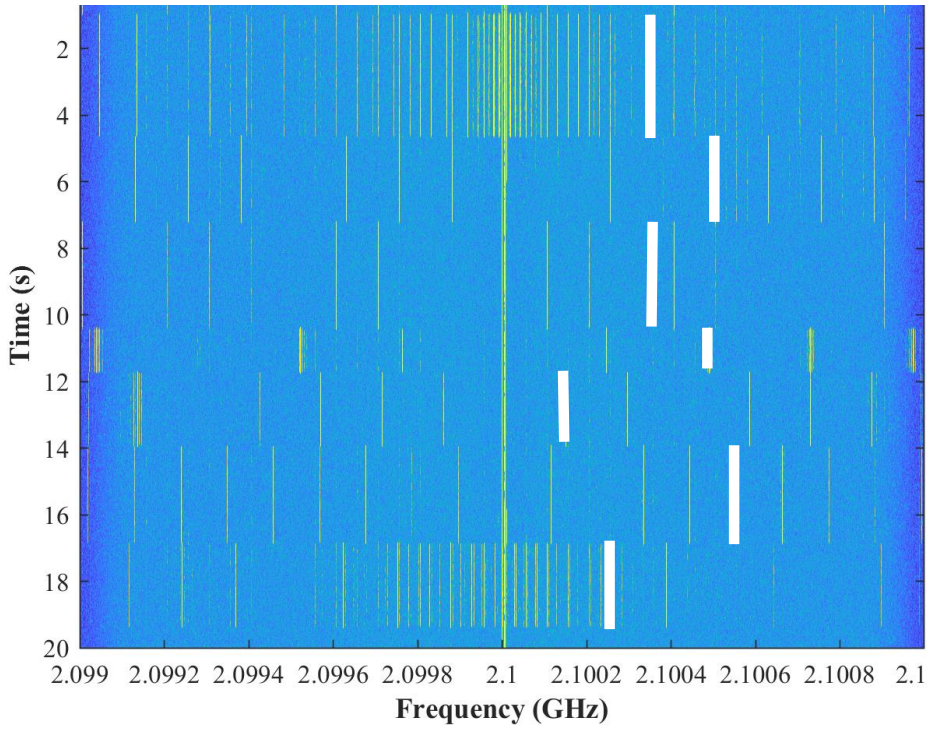


Figure 4.19: The spectrogram of the bitcount software received via the EM side-channel.

the testing setup can always be placed close to the chips under test. Therefore, for this type of application, we use near-field probes placed close to the DUT instead of horn antennas to receive stronger signals. The measurement setup for the backscattering, EM and power side-channels is shown in Fig. 4.1, which is described in Section 4.2.

To create benchmarks for the experiments, we use the T500 HT design from Trusthub, the most comprehensive HT benchmark to date [102], on an FPGA. First, we implement the Advanced Encryption Standard (AES) circuit as the host circuit, then we use the ECO (Engineering Change Order) tool in Altera's Quartus II suite to add the HT's circuitry to the host circuit without changing the placement of logic elements (and the routing of their connections) that belong to the original AES host circuit. Fig. 4.24 illustrates the placement of the HT-free circuit and the HT-afflicted circuit with a zoom-in to show the details where the HT's logic elements are placed.

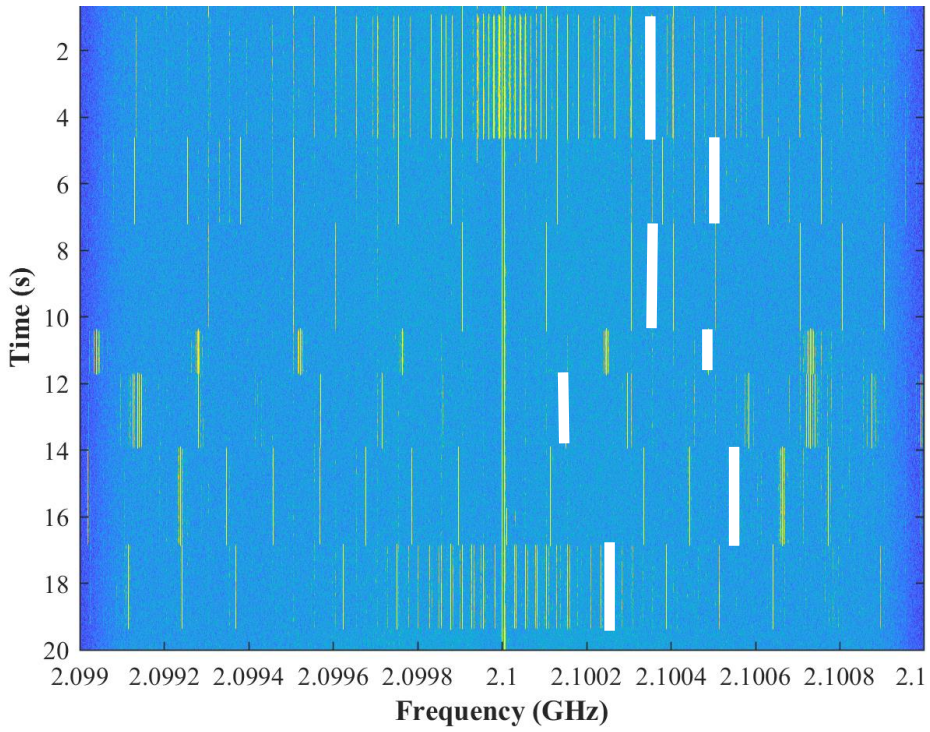


Figure 4.20: The spectrogram of bitcount software received via the power side-channel.

Evaluation

We use the setup in Fig. 4.1 (a) to record the backscattered signal. Fig. 4.25 compares the normalized amplitude ratios for the HT-free AES design to the same AES design (and layout) with the AES-T500 Trojan added. We can observe that there are a number of trace points where both sets of HT-afflicted measurements deviate significantly from HT-free measurements, and that this deviation tends to be larger for measurements in which the HT has been activated. The higher deviation from HT-free measurements seen for active-HT measurements agrees with the intuitive reasoning that a HT is easier to detect when it is active. Even so, the backscattering-based HT detection prototype successfully reports the existence in each dormant-HT experiment (100% accuracy), while reporting all HT-free measurements as HT-free (no false positives).

We repeat the same experiment as in Fig. 4.25 but measure the EM and power side-channels instead of the backscattering side-channel signal. The normalized amplitude ratios

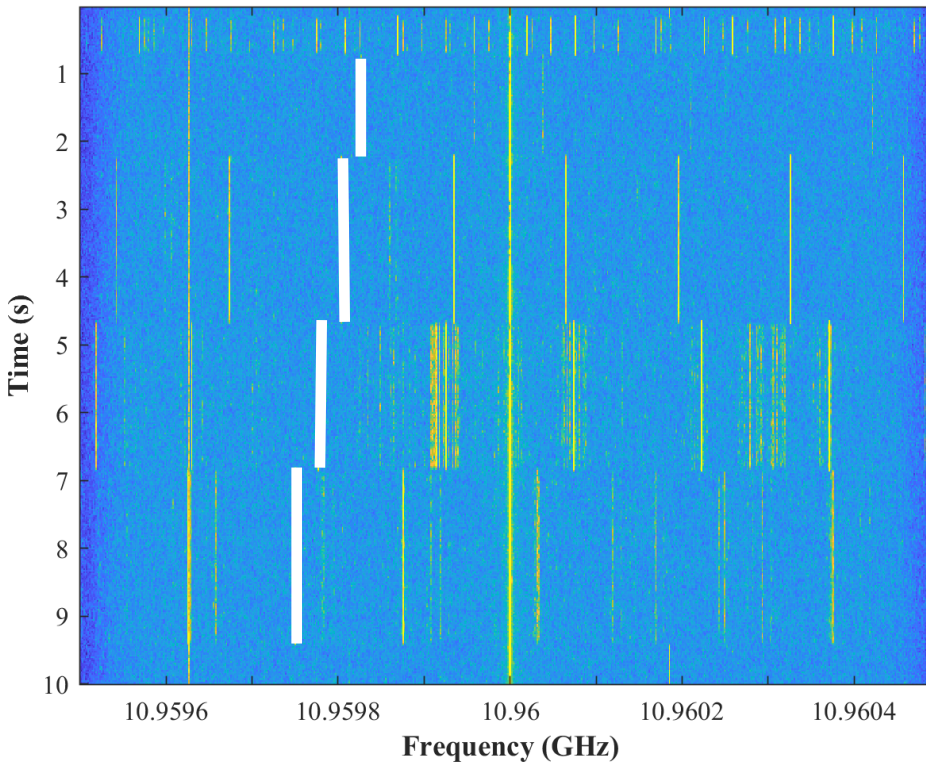


Figure 4.21: The spectrogram of basicmath received via the backscattering side-channel.

from these measurements are shown in Fig. 4.26 and Fig. 4.27. In the figures, the HT-afflicted measurements are much less separated from HT-free ones than they were using the backscattering side-channel. More importantly, nearly all dormant-HT measurements have significant overlap with HT-free measurements, making the dormant-HT measurements difficult to be distinguished from HT-free ones. This is confirmed by the receiver operating characteristic (ROC) curves shown in Fig. 4.28 which were obtained by running the same HT detection described in Section 4.4.2 on the measurements from all three side-channels. As shown in the figure, the backscattering side-channel successfully detects 100% HTs with 0% false positives. In contrast, the EM side-channel can only detect around 75% of active-HTs and only 15% of dormant-HTs. The power side-channel performs even worse, with less than 5% of HTs detected for both dormant and active HTs. This confirms the backscattering side-channel is more effective for HT detection than traditional EM and power side-channels. It is because the variation in the circuit caused by dormant HT is

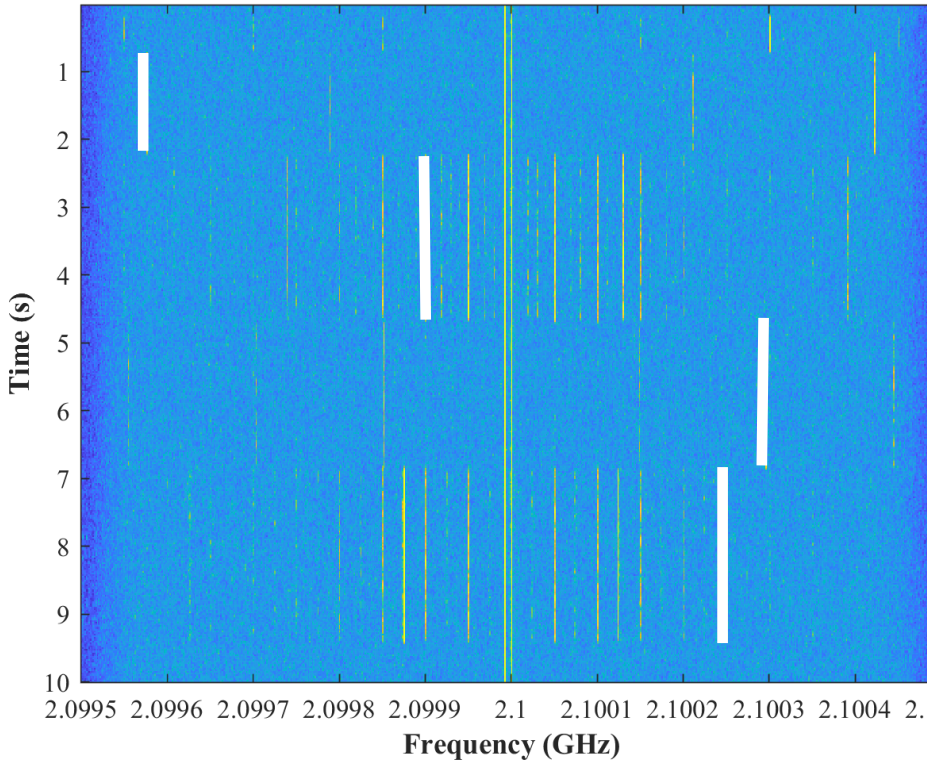


Figure 4.22: The spectrogram of the basicmath received via the EM side-channel.

relatively small. Therefore, the required sampling rate of the side-channels is very high to be able to detect HTs. As discussed, the power side-channel has a limited bandwidth. While the EM side-channel may have higher bandwidth compared to the power side-channel, it is still lower than that of the backscattering. Furthermore, its signal-to-noise ratio is affected by noise and interference, which results in poor detection accuracy compared to the backscattering side-channel.

To further evaluate and compare the performance of the three side-channels on HT detection,we implement two different circuits, PIC16F84 and RS232, and two other HT designs, PIC16F84-T300 and RS232-T100, from Trusthub [102]. The normalized amplitude ratios from the measurements for PIC16F84 circuit using the backscattering, EM, and power side-channels are shown in Figs. 4.29, 4.30 and 4.31, respectively. In the figures, the HT-afflicted measurements are much less separated from HT-free ones for the EM and power side-channels than they are for the backscattering side-channel. The ROC curves in

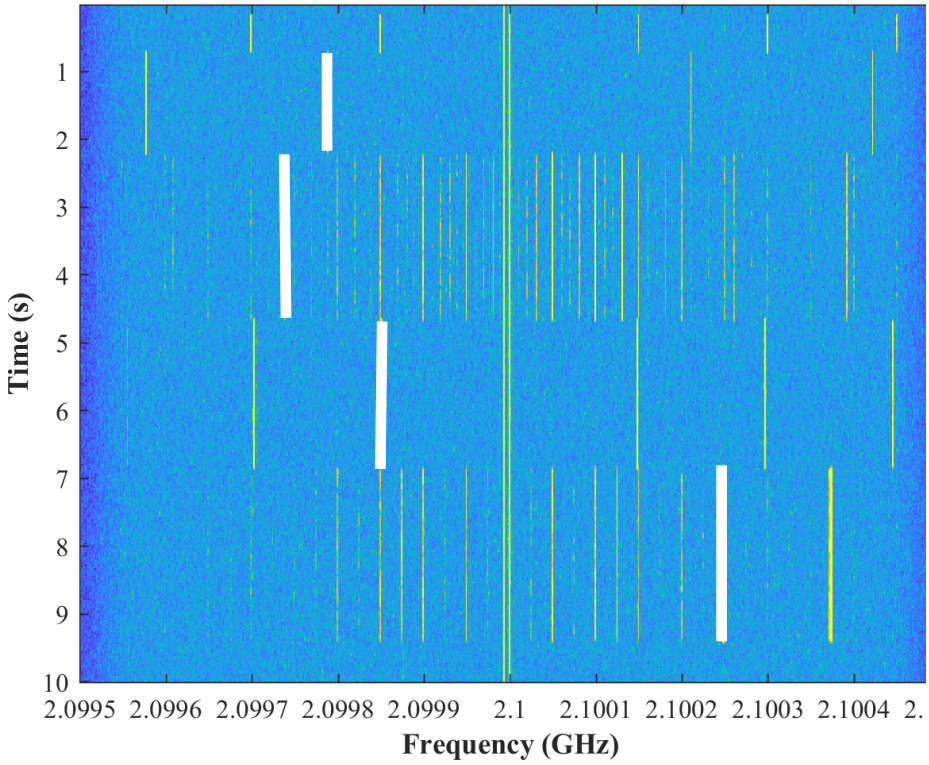


Figure 4.23: The spectrogram of basicmath received via the power side-channel.

Fig. 4.32 show that the backscattering side-channel successfully detects 100% HTs with 0% false positives while the EM and power side-channels can only detect around 30% and less than 5% of dormant PIC16F84-T300, respectively. Figs. 4.33, 4.34 and 4.35 show the normalized amplitude ratios from the measurements for RS232 circuit using backscattering, EM, and power side-channels, respectively. The ROC curves in Fig. 4.36 show that while the EM and power side-channels can only detect around 20% and 15% of dormant RS232-T100, respectively, the backscattering side-channel successfully detects 100% HTs with 0% false positives.

4.5 Conclusions

Side-channel analysis is a powerful tool both from attacker's and from defender's perspectives. Understanding similarities and differences among a large number of side-channels is a necessary step in better utilizing them. This chapter addresses this problem by mod-

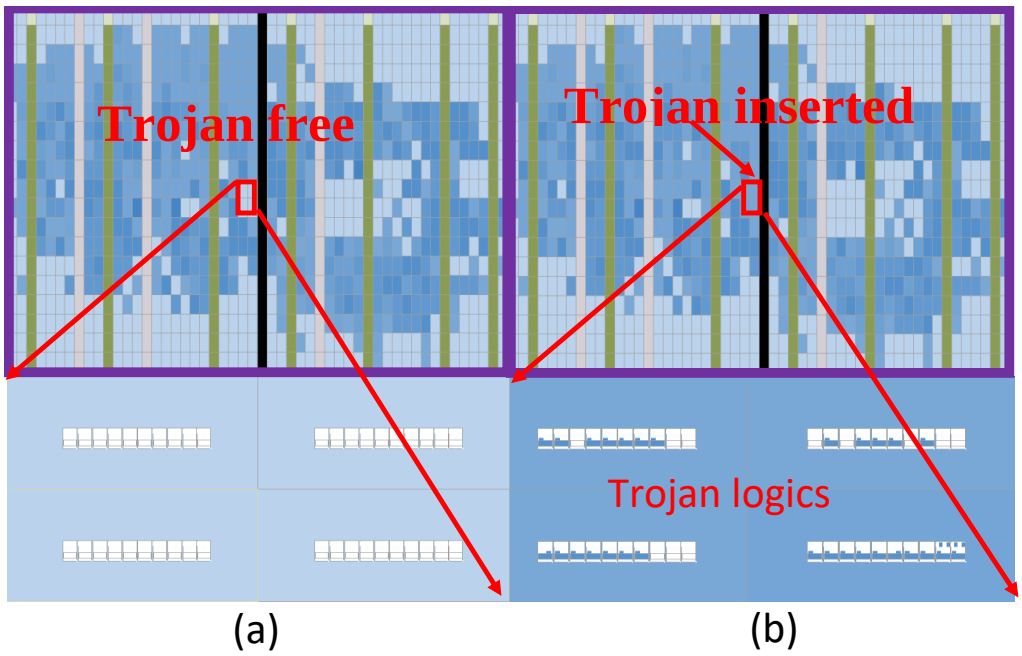


Figure 4.24: (a) Genuine AES circuit (b) Hardware Trojan infected AES circuit.

eling and quantitatively comparing the backscattering, EM, and power side-channels and discussing the performance of these three side-channels in detecting software malware and HT. The results show that for larger changes in the signals, such as those caused by malware intrusions, all three side-channels perform similarly. However, when smaller changes need to be observed, such as those caused by HTs, backscattering side-channel outperforms EM and power side-channels.

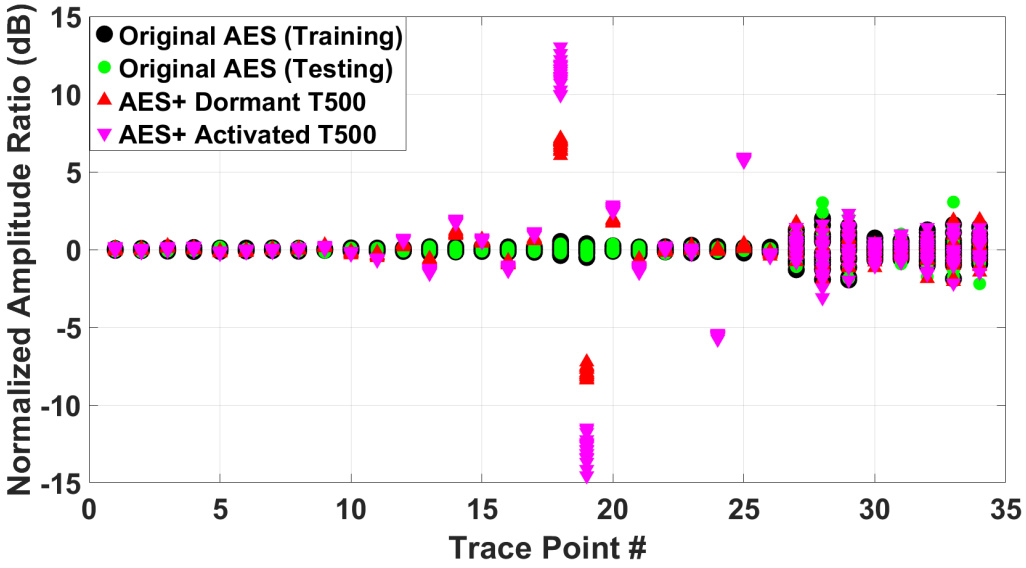


Figure 4.25: The backscattering side-channel amplitude ratios.

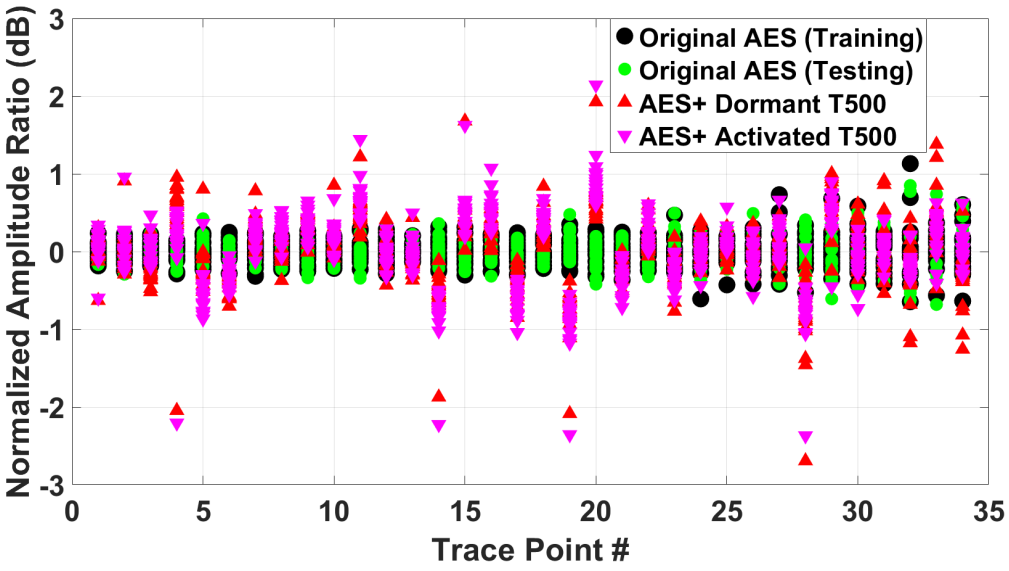


Figure 4.26: The EM side-channel amplitude ratios.

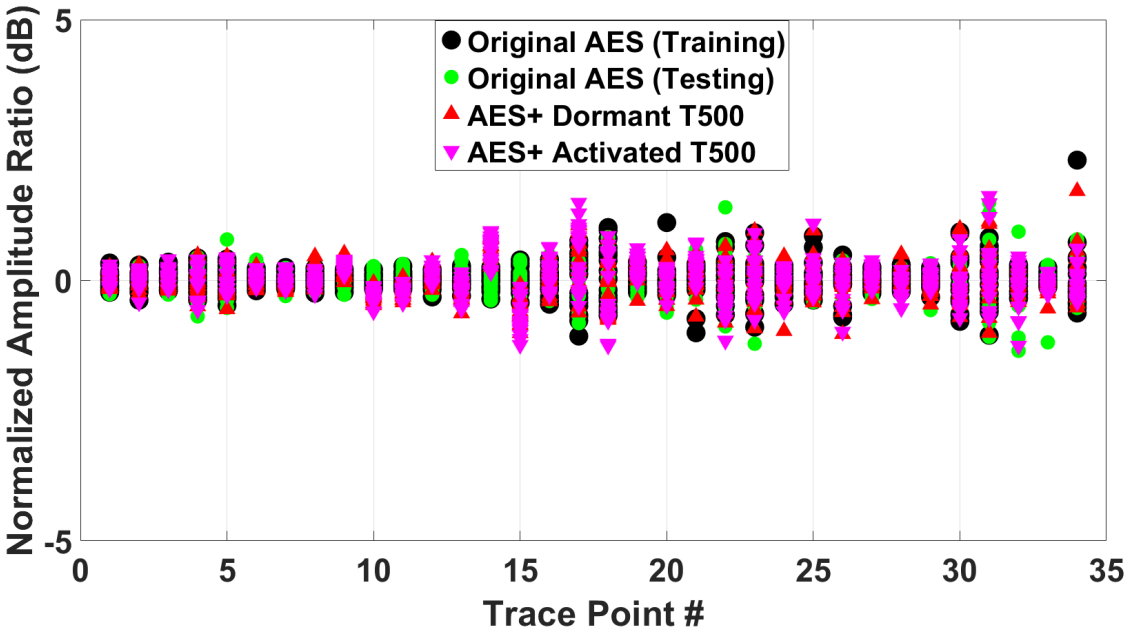


Figure 4.27: The power side-channel amplitude ratios.

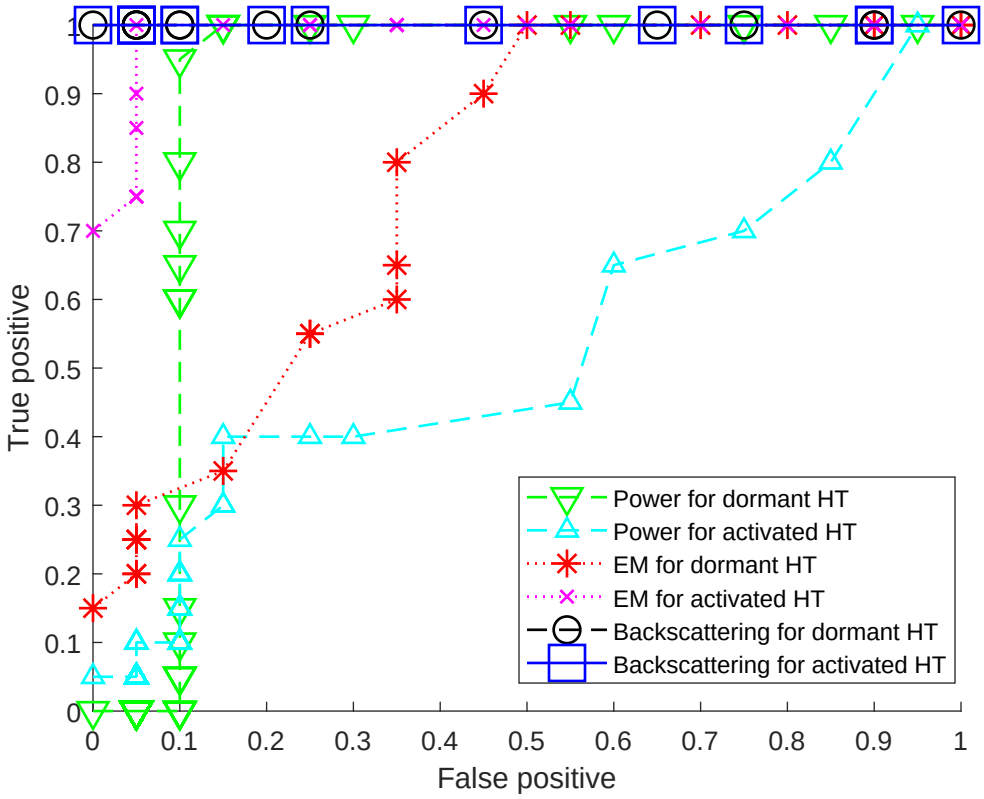


Figure 4.28: ROC curves for the backscattering, EM-based, and power-based HT detection.

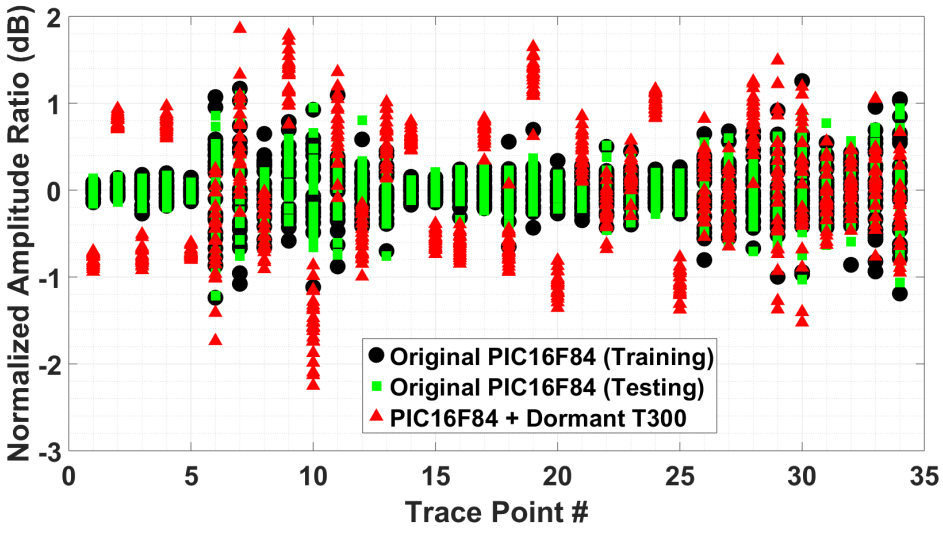


Figure 4.29: The backscattering side-channel amplitude ratios for PIC16F84.

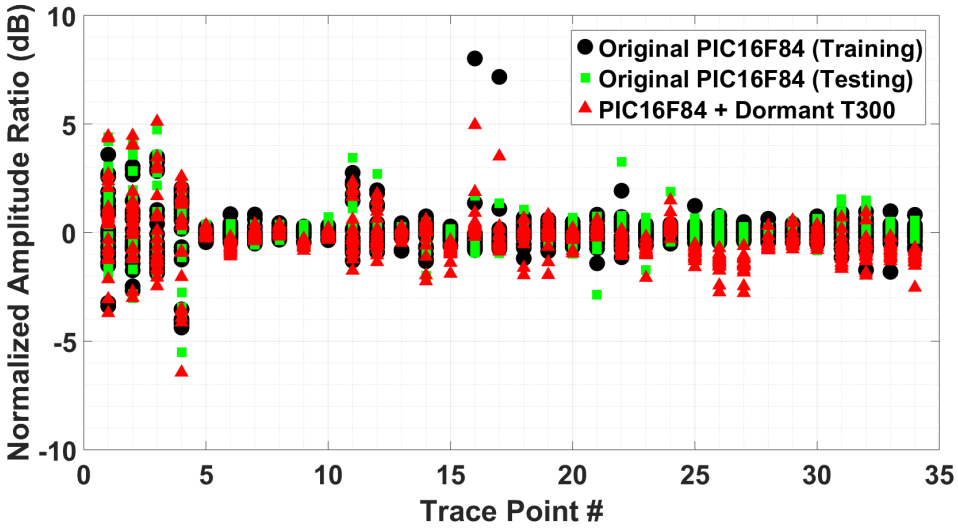


Figure 4.30: The EM side-channel amplitude ratios for PIC16F84.

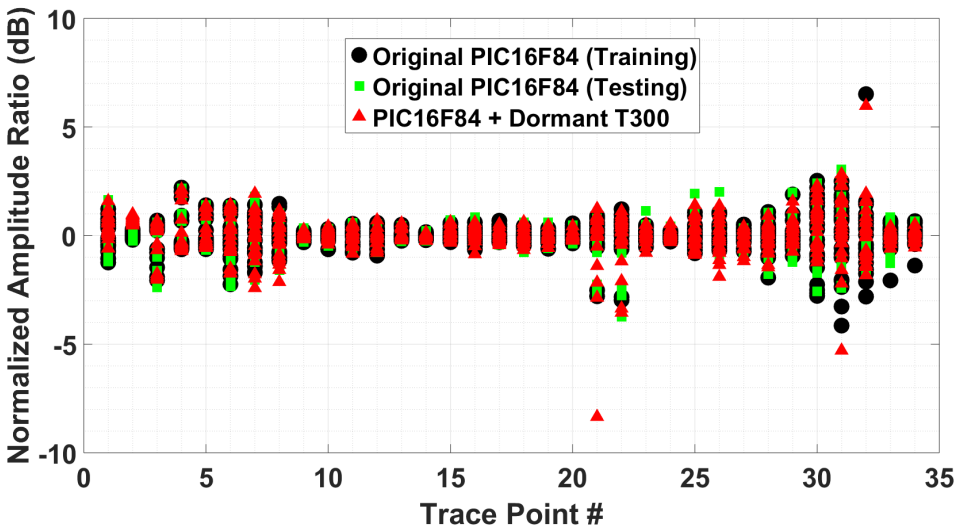


Figure 4.31: The power side-channel amplitude ratios for PIC16F84.

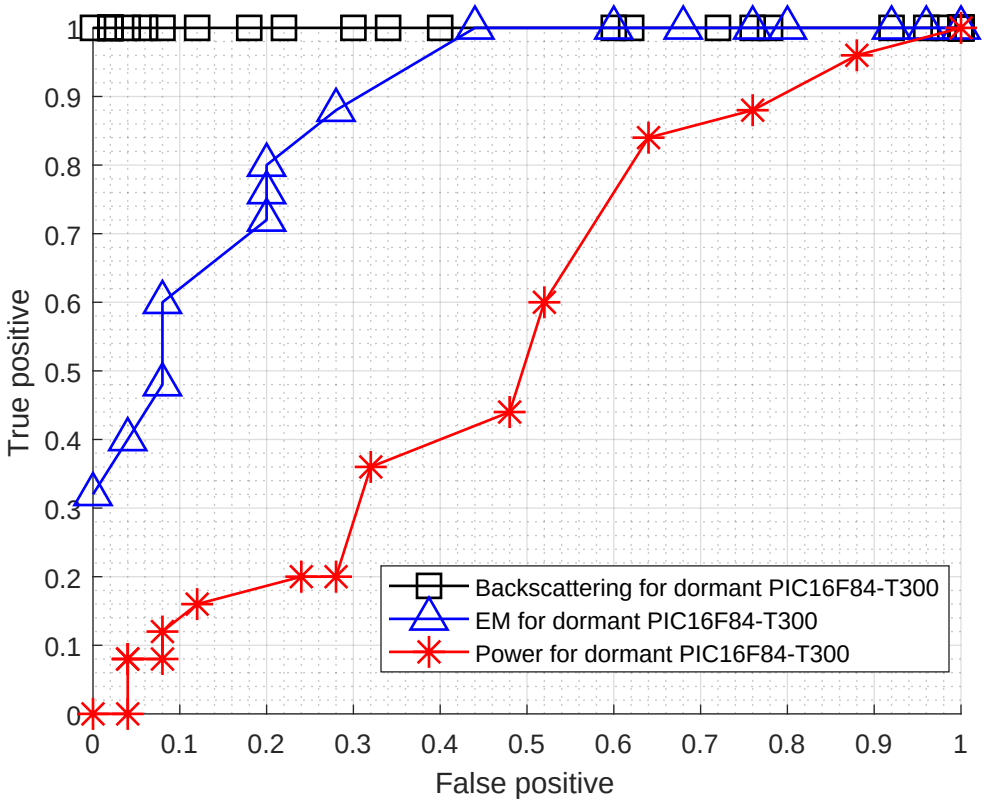


Figure 4.32: Detection performance (ROC curve) comparison of for the backscattering, EM-based, and power-based HT detection for PIC16F84.

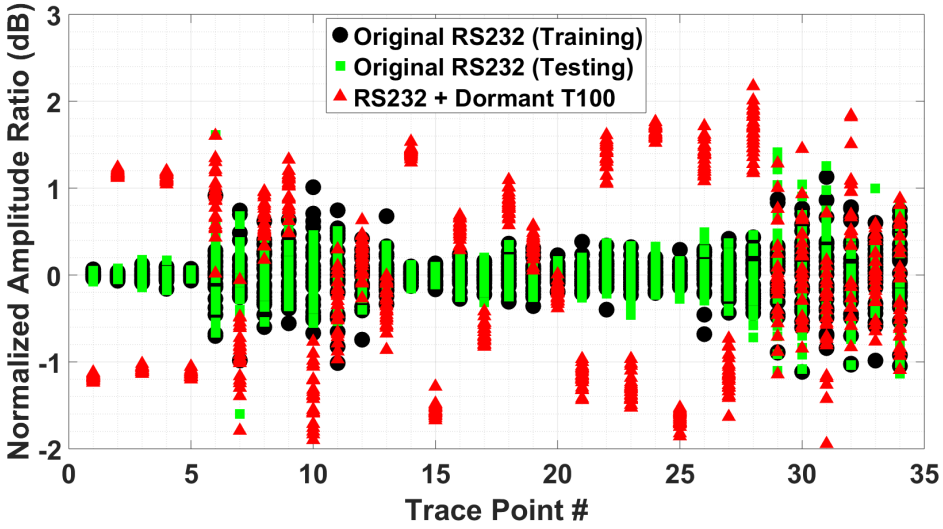


Figure 4.33: The backscattering side-channel amplitude ratios for RS232.

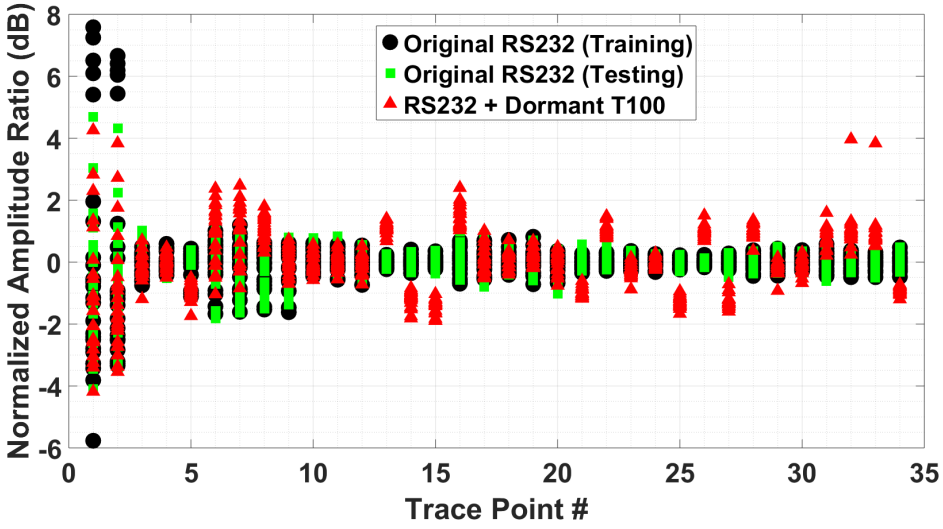


Figure 4.34: The EM side-channel amplitude ratios for RS232.

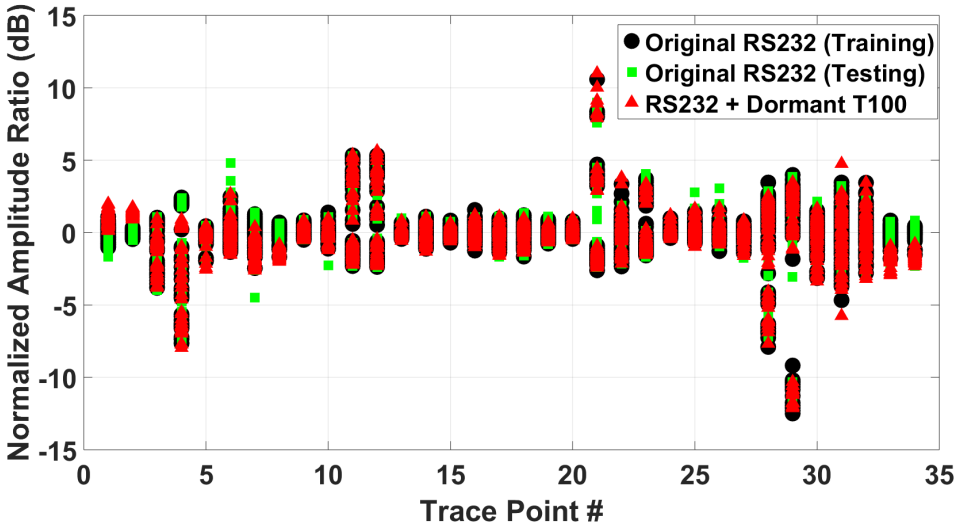


Figure 4.35: The power side-channel amplitude ratios for RS232.

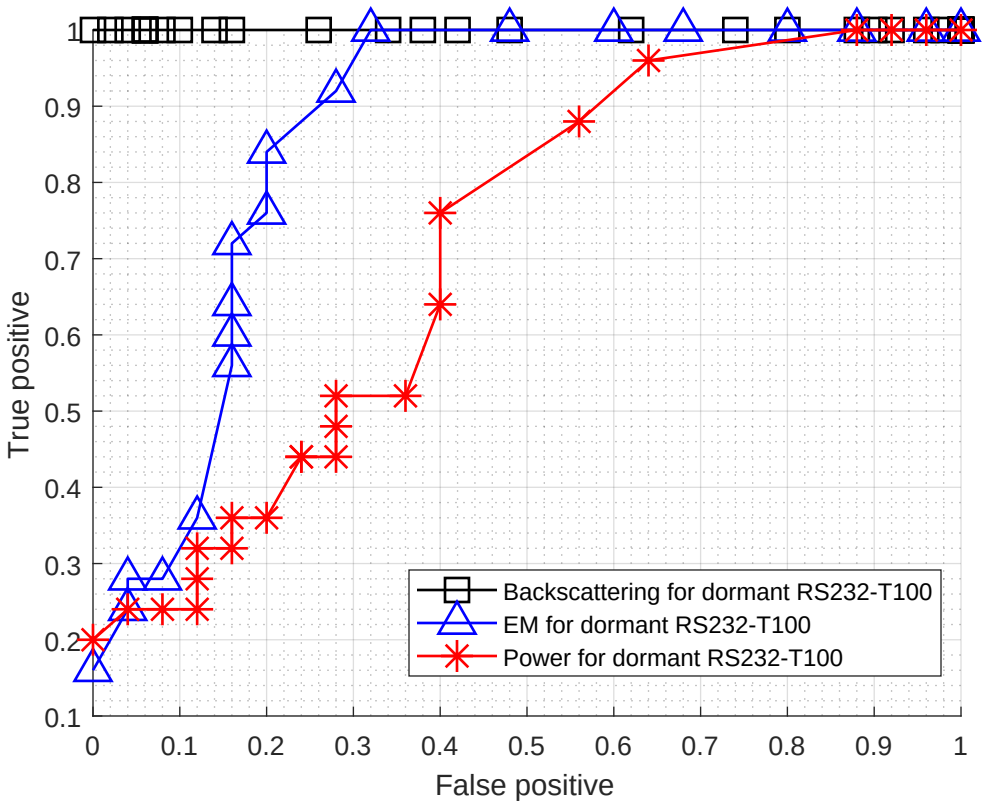


Figure 4.36: Detection performance (ROC curve) comparison of the backscattering, EM-based, and power-based HT detection for RS232.

CHAPTER 5
A NOVEL GOLDEN-CHIP-FREE CLUSTERING TECHNIQUE USING BACKSCATTERING SIDE-CHANNEL FOR HARDWARE TROJAN DETECTION

5.1 Overview

As we discussed in Chapter 1, among proposed hardware Trojan detection techniques, reverse engineering based approaches and side-channel analysis techniques are the most widely used hardware Trojan detection methods. Reverse engineering based approaches appear to be the most accurate and reliable ones because they work for all circuits and Trojan types without a golden example of the chip. However, because reverse engineering is an extremely expensive, time-consuming, and destructive process, it is exceedingly difficult for these techniques to be applied for a large population of ICs in a real test environment.

On the other hand, the side-channel based techniques can be applied to a large population of ICs because side-channel measurements do not require damaging the board while conducting testing. However, the disadvantage of side-channel techniques is their dependence on either having a "golden" (HT-free) chip, which is not a practical assumption for foundry-inserted HTs in single-source ICs, or having a detailed simulation model, which is often impractical (complex ICs, 3rd-party IP, etc.).

To overcome these shortcomings of both types of approaches, this chapter proposes a novel "golden-chip-free" clustering algorithm using backscattering side-channel. This technique is bridging the gap between expensive, destructive reverse-engineering and traditional side-channel detection techniques. The technique requires no golden chip or a priori knowledge of the chip circuitry. The proposed clustering algorithm clusters a large population of ICs based on the effect of a hypothetical HT would have on the backscat-

tering side-channel signal. In practical terms, the technique creates clusters such that the ICs in each cluster can be considered equivalent in terms of presence or absence of an HT. This allows reverse-engineering of one IC in each cluster to be used to assess the status (in terms of HT presence and nature) of that entire cluster. This significantly reduces the size of test vectors for reverse engineering based detection techniques, thus enables deployment of reverse engineering approaches to a large population of ICs in a real testing scenario.

The results are collected on 100 different FPGA boards where boards are randomly chosen to be infected or not. The results show that we can cluster the boards with 100% accuracy and demonstrate that our technique can tolerate manufacturing variations among hardware instances to cluster all the boards accurately for 9 different dormant Trojan designs on 3 different benchmark circuits from Trusthub. We have also shown that we can detect dormant Trojan designs whose trigger size has shrunk to as small as 0.19% of the original circuit with 100 % accuracy as well.

The rest of the chapter is organized as follows. Section 5.3 defines the problem and attack scenarios. Section 5.4 explains our clustering technique and algorithm, while Section 5.5 describes our experiment setup and testing scheme formulation. Section 5.6 presents the our results, while Section 5.2 discusses related work. Finally, Section 5.7 concludes the chapter.

5.2 Related Work

For the past few years as hardware Trojan have emerged as an increasingly dangerous threat and machine learning has become such a hot topic, a few HT detection techniques utilizing machine learning for clustering has been proposed. In general, machine learning can be combined with all above-discussed categories of HT detection method such as reverse engineering, functional validation, code and gate-level netlist analysis, functional testing, and side-channel analysis. In [111], the authors exploit support vector machine (SVM) and K-means clustering approach to provide automatic layout identification in their reverse

engineering-based detection method. The technique does not rely on a golden sample; however, because the nature of reverse engineering is extremely cost and time-consuming, it is not realistic to assume having a large set of data for clustering. The methods in [112, 113] propose low overhead clustering-based detection technique for runtime Trojan detection. However, the methods need golden samples for training, and is only capable of detecting activated HTs, which is not practical. The authors in [114] propose a technique using AdaBoost Meta-Learner algorithm based on automatic feature selection using Haar-like functions to assist reverse engineering detection. However, it also depends on having golden samples. There are only a few machine learning based techniques which can eliminate the need of golden samples [75, 11, 76]. However, the majority of these are pre-silicon approaches, which means that they can not detect HTs inserted in the fabrication stage

Over the past few years, a number of HT detection techniques using side-channel analysis for detecting HTs inserted in the fabrication stage have been proposed. The authors in [73] propose a novel method to detect hardware Trojans in the fabricated ICs by creating a backscattering side-channel. The results shows that their method can detect dormant hardware Trojans with 100% accuracy and 0% false positives. However, similar to the majority of other side-channel techniques, their approach requires having a verified HT-free chip. In [22], the authors present a method using EM to detect HTs without having a golden circuit by modeling the benchmark circuits they used for testing. They have simulated the models to generate EM traces for the circuit and compare them with the measured ones to detect HTs with no HT-free chip. However, in the paper, the authors only test their technique on a single FPGA board, thus the hardware manufacturing variation are not verified. Furthermore, they only evaluate their techniques with activated hardware Trojans, which is also not practical because it is extremely difficult to activate HTs without a priori knowledge of their circuitry and activation mechanisms. In addition, the technique requires a priori knowledge of the chip circuitry, heavily depends on the accuracy of the model and the simulator that generate the reference signals, and might not work for other circuits that are not

modeled in the paper.

As machine learning has become prevalent over the last decade, a number of papers exploiting clustering techniques for HT detection have been proposed. In [111], the authors exploit the support vector machine (SVM) and K-means clustering approach to provide automatic layout identification in their reverse engineering-based detection method. The technique does not rely on a golden sample; however, because the nature of reverse engineering is extremely costly and time-consuming, it is not realistic to assume having a large set of data for clustering. The methods in [112, 113] propose a low overhead clustering-based detection technique for runtime Trojan detection. However, the methods need golden samples for training and are only capable of detecting activated HTs. The authors in [114] propose a technique using the AdaBoost Meta-Learner algorithm based on automatic feature selection using Haar-like functions to assist in reverse engineering detection. However, the method also requires to have golden samples.

Only a few clustering techniques can eliminate the need for golden samples [75]-[76]. The authors in [75] present an information-theoretic approach that estimates the statistical correlation between the signals in a design and then use a weight normalization and clustering algorithm to detect HTs. In [11], the authors propose COTD, an HT detection technique based on analyses of the controllability and observability of gate-level netlist and utilizing an unsupervised clustering to detect HTs by exploiting significant inter-cluster distance caused by the controllability and observability characteristics of Trojan gates. In [76], the authors propose a technique based on "outliers", a procedure to identify suspicious signals in a netlist, and clustering technique to detect HTs. However, all of these methods are pre-silicon approaches, which means that they can not detect HTs inserted in the fabrication stage. A post-silicon clustering technique using side-channel analysis has been proposed in [77], but authors only test their method on a set of two FPGA, which does not give enough statistics to evaluate manufacturing variations among different hardware instances. One of the main challenges of techniques using side-channels with external-measurement is that

the variation across different hardware instances may cloud the difference causes by hardware Trojans. Therefore, detection accuracy normally decreases dramatically when testing across multiple hardware instances. In addition, the technique uses power side-channel, which provides very limited resolution and bandwidth [73]. As a result, the technique only gives 93.75% accuracy for HT benchmarks from Trusthub, even when testing with only two different FPGA boards.

5.3 Attack Scenarios and Problem Statement

5.3.1 Attack Scenarios

During the fabrication process at foundries, if an adversary has access to the chip layout and adds HTs to the design, a part or the entire population of ICs will be injected HTs, depending on how the ICs are produced. As a result, there are three possible scenarios:

- No adversary: There is no malicious modifications to any chip. Therefore, the entire population of ICs is HT-free.
- Partial insertion: There are malicious modifications to some of the chips. This happens when different batches of ICs are fabricated at different chronological phases of production and the attacker only inserts Trojan at one or some phases. As a result, a part of the population of ICs have Trojans, while the rest are HT-free.
- Full insertion: Malicious modification exists in all of the chips. This happens when all ICs are fabricated at once, and the attacker inserts HTs to the chip layout. As a result, the entire population of ICs will be HT-infected.

5.3.2 Problem Statement

As discussed in Section 5.1, there are two methods for the detection of HTs inserted at foundries: reverse engineering and side-channel analysis. Side-channel analysis techniques have advantages of being non-destructive and relatively fast, which are suitable for testing

a large number of ICs. However, the problem with side-channel techniques is the dependence on either 1) having a "golden" chip (a chip that is a priori known to be HT-free), which is not a practical assumption if HTs were inserted at foundries, or 2) simulation, which only works for the specific circuits modeled. These difficulties prevent these techniques from being used without other assisting techniques for HT detection in practice. In contrast, reverse engineering techniques are highly accurate and need neither simulation nor a "golden" chip, which allows them to be used for the detection of HTs without any assisting techniques. However, the problem with these techniques is that the reverse engineering process is extremely expensive, time-consuming, and destructive. Hence, these techniques could not be deployed for a large population of ICs.

To circumvent the introduced difficulties faced with the previous methods, we propose a novel clustering method using backscattering side-channel to enable the deployment of reverse engineering techniques to a large population of ICs. The problem statement is as follows: There are *M* fabricated ICs, denoted as IC_1, IC_2,...,IC_{M-1}, IC_M. Utilizing each IC, a trace of features is extracted from its backscattering side-channel signals while it operates. Each IC then can be represented as a point in a high dimensional space. These ICs can be divided into clusters based on how hardware Trojan (if existed) affects their backscattering side-channel signals. The objective of the proposed clustering algorithm is to divide all tested IC into correct clusters, so that every IC in a cluster should belong to the same type in terms of whether they are affected by HTs or not. This helps to reduce the size of test vectors tremendously for reverse engineering techniques because only one IC is required to test from each cluster.

5.4 A Novel Clustering Algorithm For Hardware Trojan Detection

5.4.1 The Impact of Hardware Trojan on Backscattering Side-Channel Signal

Nguyen et al. [73] have shown that HTs can be detected by analyzing impedance changes within sub-clock samples, where the changes caused by HTs happen and can be observed

on the clock signal. Figure 5.1 illustrates a theoretical example of a clock signal modeled as a square wave with added Gaussian noise. Figure 5.1 shows a theoretical example of a clock signal affected by HTs. As shown in the figures, if we can capture the backscattered signal of sub-clock samples where the changes caused by HT can be observed, we can detect the presence of HTs. However, the problem with the time-domain signal is that they are often very noisy, therefore, difficult to extract and synchronize measurements to get samples where changes caused by HTs happen.

In contrast, the changes caused by HTs occurring abruptly at some point in the clock cycle can be observed in frequency domain by performing short Time Fourier transformation (STFT) on time-domain signal and observe which frequency components of time domain signal are affected when dormant HT is present. Figure 5.3 shows Trojan-free and Trojan-affected clock signals in the frequency domain by taking FFT of signals given in Fig. 5.1 and Fig. 5.2, respectively. The signals in the frequency domain are much easier to measure, and the noise power is very small because of focusing a single frequency bin at a time. As a result, instead of measuring the time domain signal, we measure multiple harmonics of the clock in the frequency domain to observe changes in sub-clock samples for HT detection.

The change caused by HTs will be reflected in backscattered signals at the circuit's clock harmonics: $f_{carrier} \pm f_c$, $f_{carrier} \pm 2 * f_c$, etc. The first clock harmonic at $f_{carrier} \pm f_c$ follows the overall RCS change during a cycle, while the remaining harmonics are affected by the rapidity of change (rise/fall times), and timing of the impedance changes within the clock cycle. For each circuit, we measure the amplitude of the first *N* harmonics of the clock from its backscattering side-channel signals to form a vector, which characterizes the circuit's overall amount, timing, and duration of impedance-change activity during a clock cycle. If there is a hardware Trojan in the circuit, this vector will be different from the ones recorded from an HT-free same circuit. As a result, we can represent each circuit by a vector of *N* points, which are the amplitudes of the first *N* harmonics of the clock from its

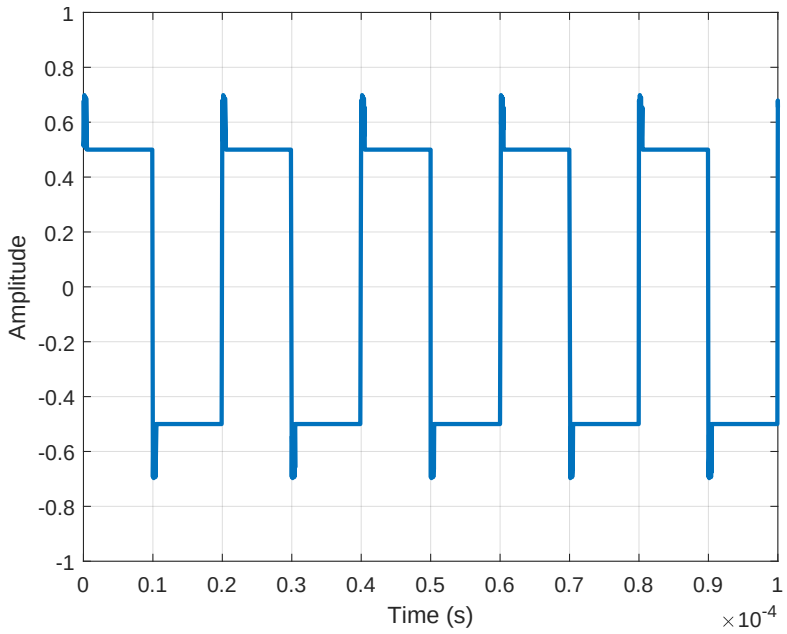


Figure 5.1: An example of a clock signal with noise.

backscattering side-channel signals: $\mathbf{h} = [h_1, h_2, ..., h_{N-1}, h_N]$, where h_j is the amplitude of the j^{th} harmonic of the clock. These vectors will be used as inputs for our clustering algorithm.

If changes caused by HTs in the time-domain signal become briefer in duration, the changes among clock harmonics become smaller in magnitude and shift to higher harmonics which, compared to lower harmonics, tend to be affected more by noise. This is one of the reasons why the backscattering side-channel works better for HT detection than other traditional analog side-channels such as EM and Power side-channels. The backscattering side-channel is a consequence of the impedance changes in switching digital switching circuits, which is caused by the transistors' two-state impedances reflecting a modulated signal. For each gate that switches, the impedance change persists for the rest of the cycle. On the other hand, the EM and power side-channels are consequences of the variation of the current flow in a circuit. As a gate switches, the current will be charged or discharged quickly, which means a current burst occurs for a very short period of time.

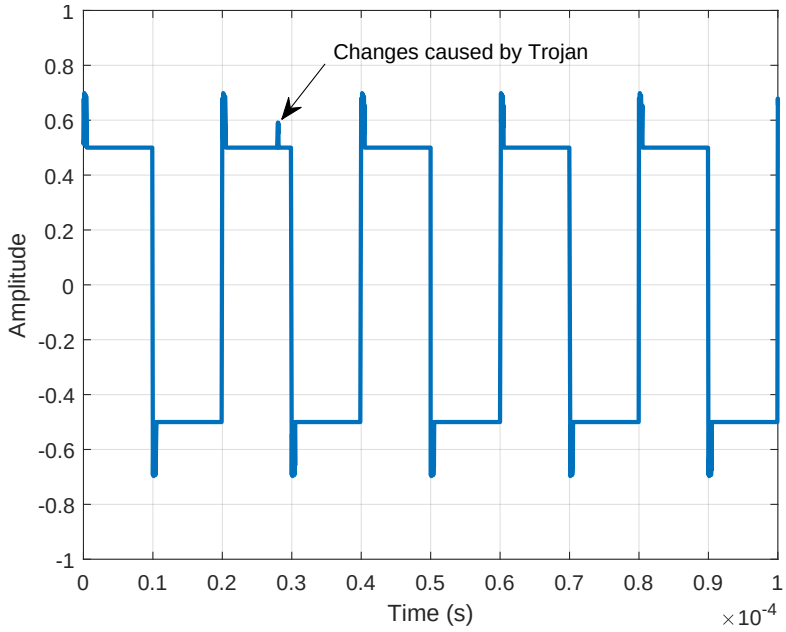


Figure 5.2: An example of a clock signal affected by hardware Trojan.

5.4.2 Graph Model for Clustering Results

This section presents the proposed methodology to categorize ICs into clusters based on how HTs (if present) affect their backscattering side-channel signals. Here we assume $\mathbf{y}_i = \begin{bmatrix} y_{i1} & y_{i2} & \cdots & y_{i(N-1)} \end{bmatrix}$ to be a vector containing the amplitude ratios of harmonics for the i^{th} board such that

$$y_{ij} = 10 * \log_{10}(\mathbf{h}_i(j+1)/\mathbf{h}_i(j)) \tag{5.1}$$

where $\mathbf{h}_i \in \Re^N$ is a vector containing the harmonic amplitudes for the i^{th} board. We use the amplitude ratio instead of the amplitude itself to cancel out the attenuation caused by the distance that affects all harmonics. We convert harmonic ratios from linear-domain to dB-domain to prevent the magnitude dominance of the top ratios, and to increase the effect of small harmonic ratios. Matrix $\mathbf{Y}$ is the matrix containing the harmonic ratios of

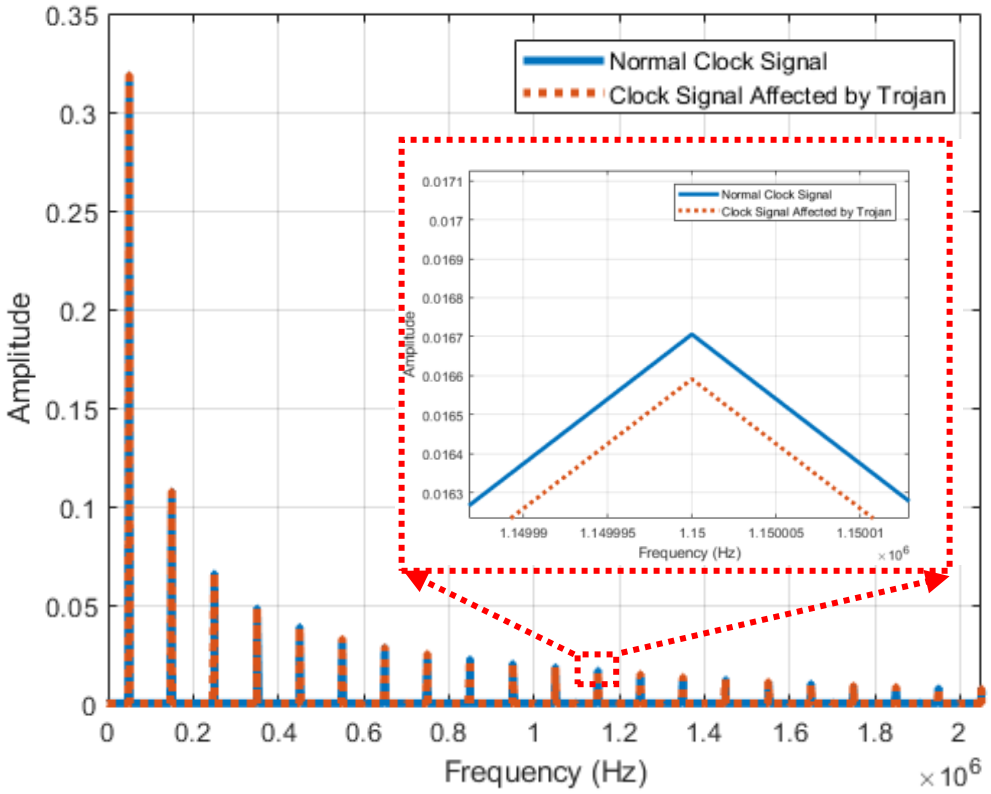


Figure 5.3: Trojan-free and Trojan-affected clock signals in frequency domain generated by fast Fourier transforming time domain signals in Fig. 5.1 and Fig. 5.2, respectively.

all boards which can be written as

$$\mathbf{Y} = \begin{bmatrix} — & \mathbf{y}_1 & — \\ — & \mathbf{y}_2 & — \\ & \vdots & \\ — & \mathbf{y}_M & — \end{bmatrix} \tag{5.2}$$

where M is the number of boards. The objective is to reveal the hidden information that could be crucial to identifying Trojans in the data by removing the redundant information. A popular technique to reduce the dimensionality of the problem is to keep the significant information by applying Principle Components Analysis (PCA). These methods are especially practical for classification when the data exhibits linear characteristics. To utilize these ideas, the first step is to obtain the singular value decomposition (SVD) of $\mathbf{Y}$ which

can be written as

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}}.$$

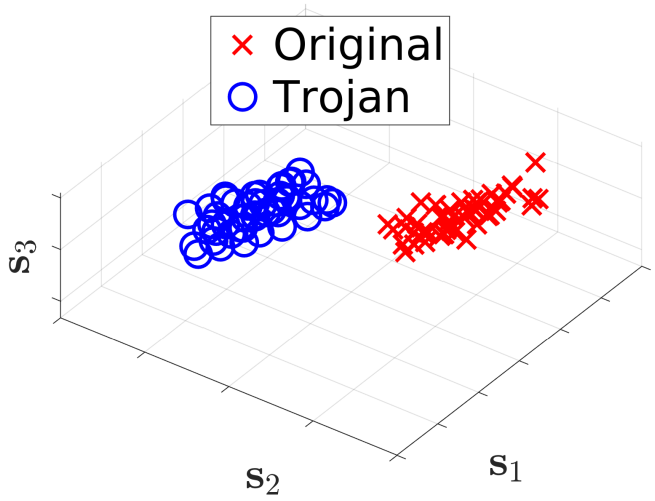


Figure 5.4: Ground truth information when half of the boards are randomly injected with a Trojan.

Here we assume that the first m singular values are the largest m singular values of the matrix $\mathbf{Y}$, and $\mathbf{V}_m$ is a sub-matrix with the first m columns of $\mathbf{V}$ corresponding to these m singular values. Therefore, to reduce the size of the data, we project $\mathbf{Y}$ onto the column space of $\mathbf{V}_m$ as

$$\mathbf{Y}_P = \mathbf{Y}\mathbf{V}_m. \tag{5.3}$$

Here, the value of m is selected so that the power of the projected data is very close to the power of $\mathbf{Y}$, i.e.,

$$\|\mathbf{Y}_P\|_F / \|\mathbf{Y}\|_F \approx 1 \tag{5.4}$$

where $\| \bullet \|_F$ is the Frobenius norm of its argument. For example, in Fig. 5.4, we plot the projected data when $m = 3$, where $\mathbf{Y}_P$ captures 99% of the power of $\mathbf{Y}$, and when half of the boards are infected with a Trojan. Here, $\mathbf{s}_j$ denotes the singular value direction corresponding to j^{th} largest singular value.

After discarding the redundant information, the next step is to find the clusters in the data. The expectation is that each cluster corresponds to different board groups due to pro-

duction variability, or existence of a Trojan. To find the clusters and corresponding centroid points, we utilize k-means algorithm. The algorithm requires the number of expected clusters, N_C, and their initial locations, $\mathbf{L}_C \in \Re e^{N_C \times m}$ (Each row represents the location of the corresponding cluster), as input. A careful selection of the initial cluster locations is important to avoid algorithm to converge to a local optimum. In that regard, we apply the following procedure to initiate the k-means algorithm:

1.) Choose a random sample from the projected data as the location of the first cluster.

2.) Find a sample whose total distance is the furthest away from the previously chosen clusters.

3.) Repeat until all centroids are initialized.

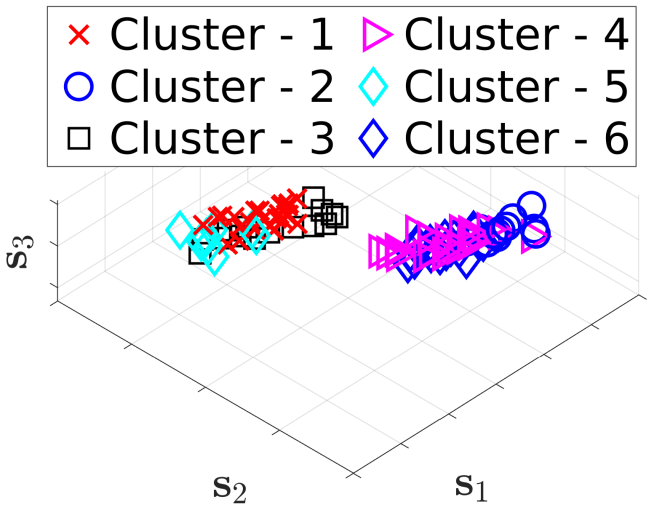


Figure 5.5: K-means clustering of the boards when the number of center points is chosen to be six.

The procedure ensures wide separation of the centroids. We need to note here that, N_C is assumed to be larger than actual number of clusters in the data, i.e., larger than the number of Trojan types. The assumption follows the fact that we have no information on how many types of Trojan may exist in the testing devices in a realistic scenario. However, having more than the number of actual clusters can be misleading because it can raise

suspicion even when there is no Trojan-affected board in the sample space. For example, in Fig. 5.5, we plot the results of the algorithm when $N_C = 6$ for the data given in Fig. 5.4. Comparing the actual labels given in Fig. 5.4, we observe that there is no cluster that contains both the original and Trojan-affected circuits. Therefore, we require a method that decreases the number of clusters to reveal the existence of Trojan-affected circuits more reliably.

To decrease the number of clusters, we propose to use graph method and the shortest path algorithm. To accomplish that, we create a graph where each arc represents that two centroids at the edges of an arc belong to the same group. Please note that "group" indicates the Trojan type or whether the board is Trojan-affected. Our proposition is that the group of two closest clusters are the same if the distance of these clusters are below some threshold. In other words, the constraint on arcs is that an arc is valid only if the distance between the cluster centroids at the edges is small than a given threshold. In that respect, the first step is to obtain a threshold automatically. We can summarize the process of choosing the threshold as follows:

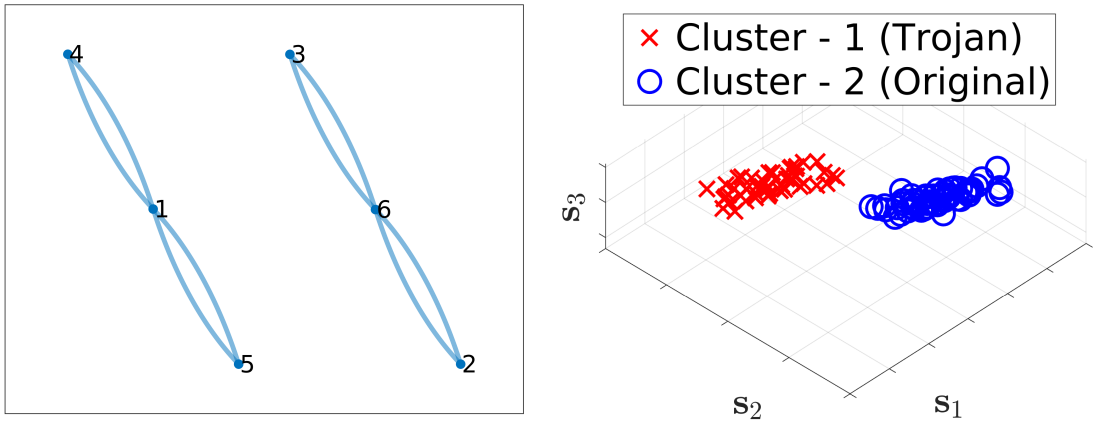
1.) Calculate the distance among centroids.

2.) Choose the closest two clusters for each cluster, and keep the distances in a list.

3.) Assign threshold as the mean distance of this list.

To illustrate how algorithm works, the graph created by the algorithm is shown in Fig. 5.6 (a) for the clusters in Fig. 5.5. The nodes corresponding to the same classes are connected. After generating the graph and identifying the valid arcs, the final step is to check whether a node is reachable from another nodes. If there exists a path between any two nodes, we label these as the same type, otherwise, we decide that the sample space contains at least two clusters, therefore, some boards are Trojan-affected. To obtain the connected nodes automatically, we exploit the shortest path algorithm [107] to check whether a node, i.e., a cluster, is reachable from another node. The algorithm returns null if there is no

path between two given nodes, and a path if these two nodes are reachable. Based on the outcome of the shortest path, we relabel the sample space indicating whether the connected nodes belong to same kind. An example of the process is given in Fig. 5.6 (b). We observe that although the exact identity of these classes are not known, it is possible to divide data into two groups, and therefore, to determine that the batch contains two circuit designs that are not identical, or some of the boards are Trojan-affected.



(a) (b)

Figure 5.6: a) Generation of the graph based on the distances between the centroids of the clusters, b) Clustering the data into two groups as Trojan injected vs. no-Trojan-free boards. Labels inside the parenthesis indicate the ground truth.

5.5 Experimental Setup and Testing Scheme Formulation

5.5.1 Experiment Setup

The experimental setup to evaluate the performance of the proposed algorithm is shown in Fig. 5.7. The setup includes a transmitter Aaronia E1 electric-field near-field probe [103] connected to an Agilent MXG N5183A signal generator [108], and a receiver Aaronia H2 magnetic field near-field probe [103] connected to an Agilent MXA N9020A spectrum analyzer [109]. The devices-under-test (DuT) are Altera DE0 Cyclone V FPGA boards [110]. An angle ruler is used as a positioner so that different DE0-CV boards can be tested using approximately the same position of probes. A laptop is used to control the devices

and automate the measurements. A 3 GHz continuous sinusoid signal is generated by the signal generator, and backscattered signals are recorded by the spectrum analyzer.

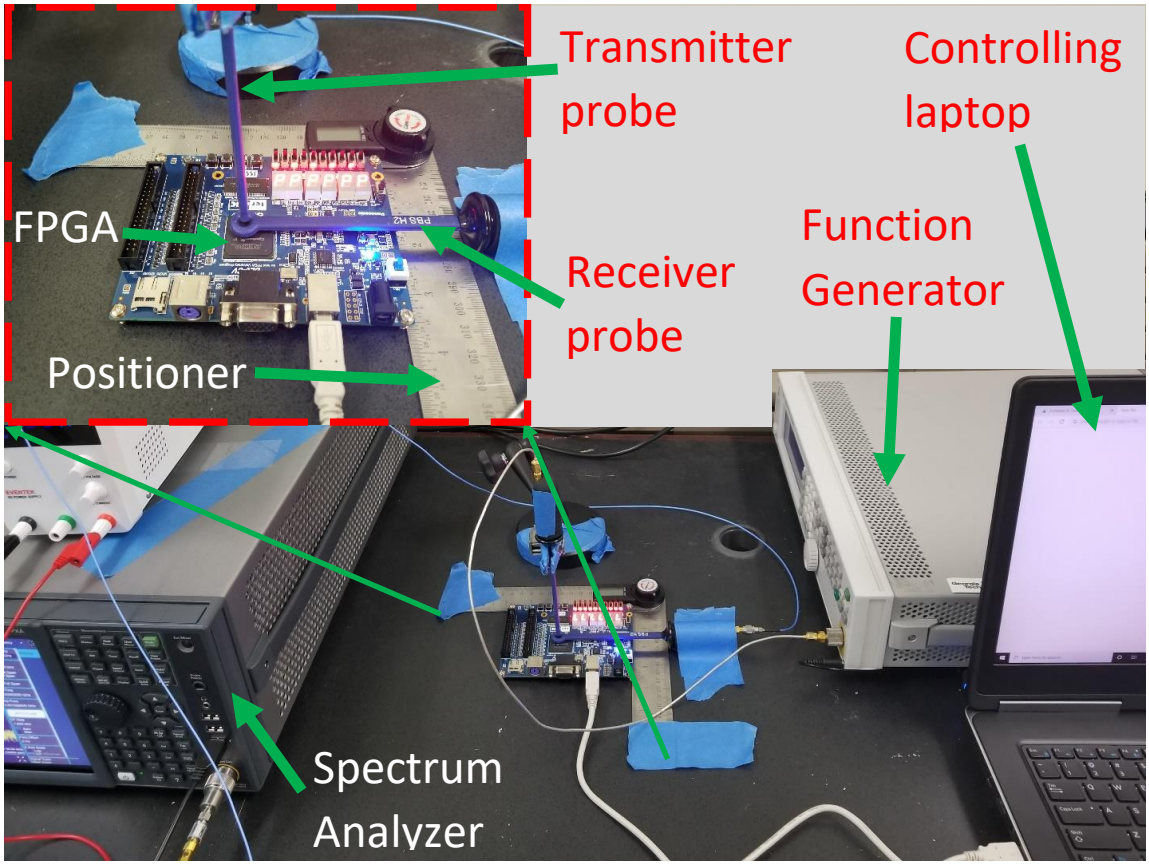


Figure 5.7: Measurement setup for IC clustering using backscattering side-channel collection for HT detection.

5.5.2 Hardware Trojan Benchmark Implementation

To evaluate our technique, we implement three different benchmark circuits AES, RS232, and PIC16F84 from the TrustHUB Trojan repository [102]. There are total of 21 Trojan designs for AES circuit, 4 Trojan designs for PIC16F84 circuit, and 21 Trojan designs for RS232 circuit. Because numerous HTs in the TrustHub repository are similar to each other, we select circuits that exhibit different approaches for their triggers and payloads. Each of these Trojans has a different triggering mechanism such as observing a specific sequence of the input, counting number of encryption rounds, observing the number of execution of a

Table 5.1: Hardware Trojan Benchmarks and Detection Results for AES, PIC16F84, and RS232 Circuits

Benchmark	**Size of Trojan (Percentage of HT-free circuit)**		
	Trigger	Payload	Total
AES-T1200	0.32%	1.61%	1.93%
AES-T500	0.28%	1.51%	1.79%
AES-T700	0.27%	1.76%	2.03%
PIC16F84-T100	1.34%	1.81%	3.15%
PIC16F84-T300	1.37%	1.96%	3.33%
PIC16F84-T400	1.35%	1.75%	3.10%
RS232-T300	1.47%	1.58%	3.05%
RS232-T600	1.50%	1.48%	2.98%
RS232-T901	1.53%	1.61%	3.11%

specific instruction, etc., and performs a different payload functionality such as shortening the hardware lifetime, leaking private keys, changing the address to program memory, etc. Table 5.1 summarizes the benchmarks we use.

The Trojan-affected and Trojan-free designs are carefully mapped to the FPGA by using ECO (Engineering Change Order) tools so that they have the same layout except for the Trojan part, thus making for a fair comparison. As mentioned in Section 2.1 in Chapter 2, it is extremely hard to activate an HT without a priori knowledge of its triggering circuit, it is highly desirable for an HT detection technique to be able to detect HT when it is dormant. As a result, our evaluation focuses on evaluating our algorithm for dormant HTs. In other words, all Trojans stay inactive in all our experiments.

5.5.3 Testing Scheme Formulation

All HT benchmarks are implemented on Altera DE0 Cyclone V FPGA, and we test 100 boards by randomly infecting the boards. To prototype a real testing environment, for each HT benchmark, we randomly program each of the 100 boards with HT-free or HT-

infected designs and record its backscattering side-channel signals while the board is running. For each board, we extract the amplitude of the first 40 harmonics of the clock from its backscattering side-channel signal. We only use 40 harmonics because the higher harmonics are very weak and shrunk under the noise floor. As a result, for each hardware Trojan benchmark, we will have a set of 100 traces, in which each trace contains 40 points, denoted as follow: $\mathbf{h}_i = [h_{i1}, h_{i1}, ..., h_{iN-1}, h_{iN}]$, where $N = 40$, and $1 \leq i \leq 100$. Our clustering algorithm takes these traces as inputs to cluster the ICs.

5.6 Evaluation

5.6.1 Evaluation of Existing Hardware Trojan Benchmarks

In this section, we provide the experimental results for Trojan detection. The process can be summarized as follows:

- $\rightarrow$ Collect the data from all boards with the setup given in Fig. 5.7. The number of boards tested for the experiments is 100.
- $\rightarrow$ Take the ratios of the consecutive harmonics, and convert them into dB-domain.
- $\rightarrow$ Collect the harmonic ratios for all boards in a matrix to generate $\mathbf{Y}$.
- $\rightarrow$ Obtain SVD of $\mathbf{Y}$, and project it into the space defined by the right-singular vectors corresponding to largest m singular values to generate $\mathbf{Y}_P$. Here, m is chosen such that it is the smallest number of singular values satisfying the following equation:

$$\|\mathbf{Y}_P\|_F / \|\mathbf{Y}\|_F \approx 0.999. \tag{5.5}$$

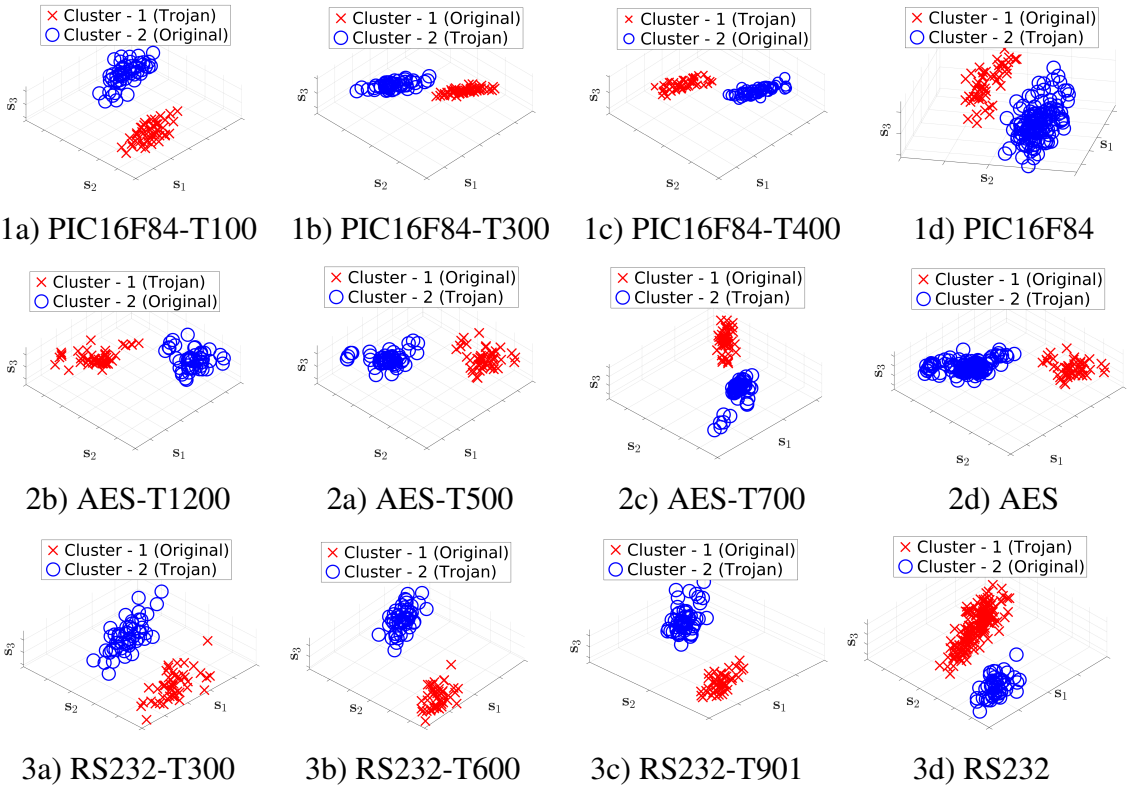
- $\rightarrow$ Apply the k-means algorithm by ensuring N_C is larger than the number of possible Trojan types. The initialization of the centroids are done based on the procedure given in Section 5.2.

→ Generate the graph of similarity with respect to the threshold calculated in Section 5.2.

→ Apply shortest path algorithm to reveal possible classes in the sample space. If the algorithm returns more than one cluster, the batch of boards contains some Trojan-affected boards.

Since the goal of the paper is to separate the Trojan-free designs from all other Trojan-affected designs, we define the accuracy of the measurements as

$$\texttt{accuracy}\ (\%) = \frac{\begin{array}{c}\text{\# of correct labeling} \\ \text{whether the design is original}\end{array}}{\text{\# of measurements}} \times 100. \tag{5.6}$$



1a) PIC16F84-T100 1b) PIC16F84-T300 1c) PIC16F84-T400 1d) PIC16F84

2b) AES-T1200 2a) AES-T500 2c) AES-T700 2d) AES

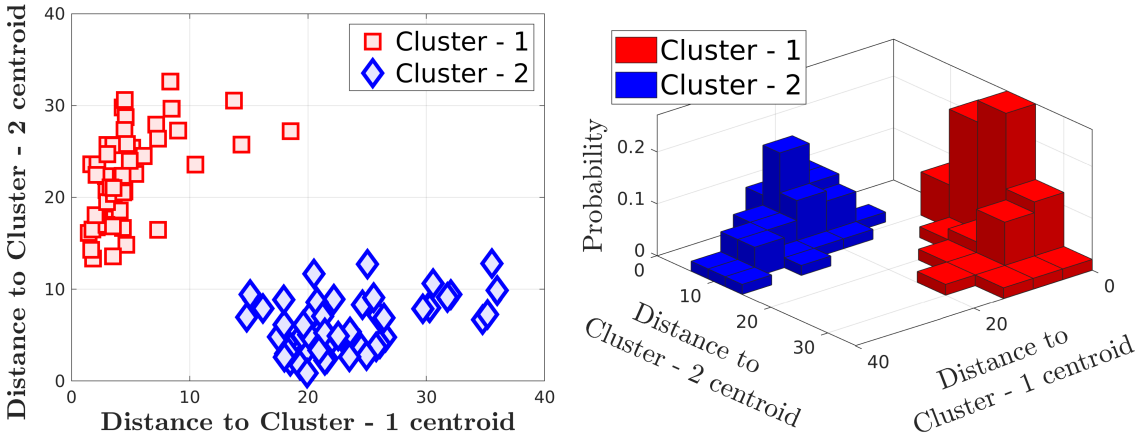
3a) RS232-T300 3b) RS232-T600 3c) RS232-T901 3d) RS232

Figure 5.8: Separation of the Trojan-free and the Trojan-affected circuits. First three columns contain the plots when only one Trojan exists, and the last column of figures are when all considered Trojans exist in the sample space.

Please note that the actual labels of the circuits are only required to calculate the accuracy of the proposed method. Therefore, after having the outcome of the procedure given above, we first identify the group which contains the most of the original designs, and then label this group as the Trojan-free and the rest as the Trojan-affected circuits. Finally, we compare our labels with the actual labels to calculate the accuracy. If the proposed method classifies all the original designs in a cluster, and if this cluster does not contain any samples from Trojan-affected designs, the accuracy of the algorithm will be equivalent to 100%.

The tested designs are given in Table 5.1. We first work on PIC16F84 circuit with 3 different Trojan designs. The results are plotted by considering the singular vectors corresponding largest three singular values. The outcome of the procedure is given in Fig. 5.8 (1a-1d). The figures in Fig. 5.8 (1a-1c) correspond to the scenarios when the batch contains only one Trojan type. However, Fig. 5.8 (1d) includes samples from all Trojan designs. The number of singular values used for these experiments that satisfies the condition given in (5.5) is 10, and $N_C = 6$. We also plot the sample distances to each cluster centroid in Fig. 5.9 (a) and their distribution in Fig. 5.9 (b) for the samples given in Fig. 5.8 (1a). The mean distances of Cluster - 1 samples to the centroids are 4.96 and 22.27 with standard deviations 3.47 and 5.03, whereas mean distances of Cluster - 2 samples are 23.39 and 6.08 with standard deviations 5.46 and 2.95, respectively. We achieve **100%** accuracy for all of the experiments. We need to note here that the legends of the figures do not give any information whether the group is Trojan-affected or original. They only provide the information that the sample space contains two different groups, hence, one of these groups represents the designs with Trojan. However, we provide the actual labels of the classes in parentheses for a better illustration.

The other experiments are done with AES and RS232 circuits. Similarly, the results are shown in Fig. 5.8 (2a-2d) and Fig. 5.8 (3a-3d) for AES and RS232, respectively. The plots in Fig. 5.8 (2a-2c) and in Fig. 5.8 (3a-3c) correspond the experiments when the board batch contains only one Trojan design type for AES and RS232, respectively. The experiments



(a) (b)

Figure 5.9: a) Distances of each circuit to the cluster centroids. b) Distribution of distances of each circuit to each cluster centroid.

with all considered Trojan designs are shown in Fig. 5.8 (2d) and in Fig. 5.8 (3d). We keep the number of clusters, N_C, same for PIC16F84 circuit. This time, the number of singular-values satisfying the equation given in (5.5) corresponds to 12 for each circuit. Similarly, we obtain **100%** accuracy for all these experiments meaning that all the original circuits are separated from the designs that is Trojan-affected, and clustered in a single group.

From the results, we can make the following observations:

I) The backscattering side-channel is a powerful mechanism to detect the existence of a Trojan when the ratios of the harmonics are exploited since the separation between the Trojan-free and Trojan-affected circuits are significant.

II) The proposed methodology (backscattered signal plus PCA and k-means algorithm) enables perfect clustering of the Trojan-free and Trojan-affected circuits.

III) When multi-Trojan designs are considered, they still behave like a single group, and the proposed method can successfully distinguish the existence of at least two different classes.

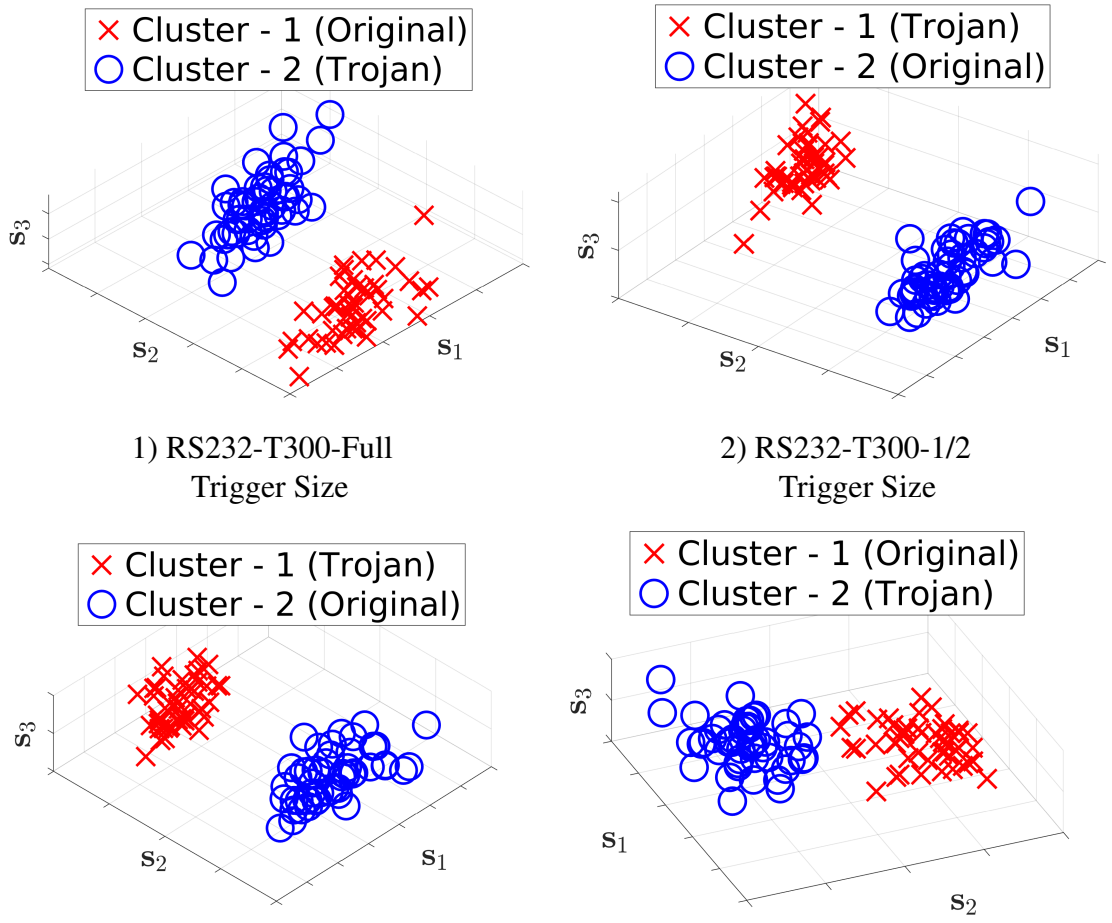
Table 5.2: Hardware Trojan Benchmarks and Detection Results for Different Size of Trojan's Trigger

Benchmark	**Size of Trojan's Trigger (Percentage of HT-free circuit)**
RS232-T300 w/ 1/2 Trigger Size	0.76%
RS232-T300 w/ 1/4 Trigger Size	0.39%
RS232-T300 w/ 1/8 Trigger Size	0.19%

5.6.2 Evaluation of Changing Size of Hardware Trojan Triggers

Because the algorithm performs so well on the existing HT designs in Table 5.1, this section focuses on testing the limit of our algorithm by reducing the size of HTs. The authors in [73] demonstrated that only the trigger is active while the payload stays inert when hardware Trojans are dormant, thus if the trigger is big enough, the Trojans can be detected regardless of its payload size. Therefore, we will focus on changing the size of the trigger to test the limits of the proposed algorithm. The RS232-T300 is chosen for this experiment because the trigger of the Trojan can be meaningfully resized. We change the size of the trigger of RS232-T300 while keeping its payload the same to create test designs that are summarized in Table 5.2.

The first goal is to investigate whether the proposed method still works when only one HT benchmark exists in the board batch. The same parameters with the experiments given in Section 5.6.1 are used for the number of clusters and singular vectors. The clustering results are given in Fig. 5.10. We again obtain **100%** accuracy in terms of separating the original circuits from the Trojan-affected ones. Here, one important observation is that as the size of the Trojan trigger decreases, the distance between centroids of the two classes decreases, i.e. the Trojan does become more similar to the original circuit when it only has 1/8 trigger size than when it has a full-size trigger. To illustrate this, we show clustering results where the measurements from all trigger sizes were included, i.e. five different designs, one HT-free and four variants of an HT-infected design (with diferent



1) RS232-T300-Full Trigger Size

2) RS232-T300-1/2 Trigger Size

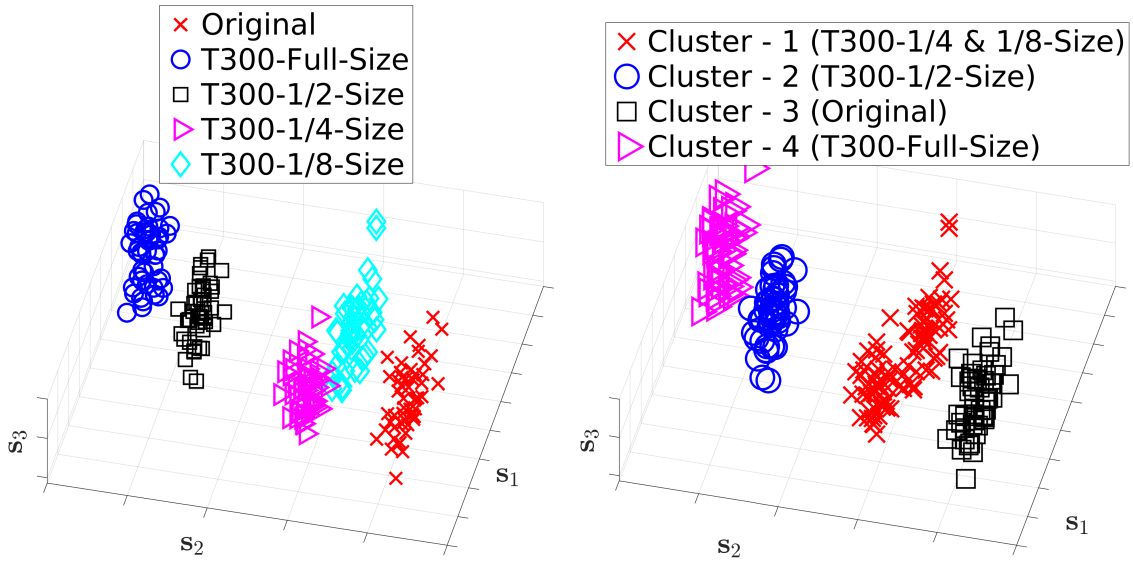
3) RS232-T300-1/4 Trigger Size

4) RS232-T300-1/8 Trigger Size

Figure 5.10: Separation of the Trojan-free and the Trojan-affected circuits when the size of RS232-T300 varies.

trigger sizes), are subjected to our clustering technique. The results are shown in Fig. 5.11, with actual (ground-truth) labels (left) and with clustering-produced labels (right). In terms of separating HT-free from HT-infected designs, the accuracy of this clustering is still **100%** (all HT-free instances are in one cluster while all HT-infected instances are in other clusters). Furthermore, the technique is able to distinguish (put in separate clusters) different variants of the HT, except for the variants with 1/4 and 1/8 triggers, which are in the same cluster. We note that the technique is able to distinguish the 1/8-trigger variant from an HT-free design, even though it did not distinguish 1/4- from the 1/8-trigger variant (the difference among them is also 1/8 of the full trigger). This is because the additional

trigger activity in the 1/4 variant is similar to the trigger activity in the 1/4 variant, i.e. it is only a matter of *how much* trigger activity the design has. In contrast, the HT-free design has no trigger activity at all, so the presence of trigger activity in the 1/8 design allows it to be well-separated from the HT-free design. This implies that HTs whose circuitry and activity mimics that of the original design would be more difficult to detect, but only up to a point – even such activity-mimicking HTs would be detected is they are sufficiently large (in this particular experiment, larger than 0.19% of the original circuit).



1) RS232-T300 Ground Truth 2) RS232-T300 Estimates

Figure 5.11: Separation of original and Trojan-affected circuits when the size of RS232-T300 varies. The experiments are performed with original, full-Trigger-size, 1/2-Trigger-size, 1/4-Trigger-size, and 1/8-Trigger-size circuits.

Based on the results given in this section and Section 5.6.1, our main observation is that our technique successfully separates HT-free from HT-infected designs, even for very small HTs (0.19% of the original circuit, in our experiments). Additionally, the technique successfully separates different HT designs from each other, except when the HTs only differ in size (but not nature) of their trigger circuitry, and that difference in size is very small (0.19% of the original circuit, in our experiments).

5.7 Conclusions

This chapter proposes a novel golden-chip-free method for clustering fabricated integrated circuits into groups for deployment of reverse engineering based hardware Trojan detection techniques to a large population of ICs. Our technique classifies boards into clusters based on how hardware Trojans (if existed) affect their backscattering side-channel signals. Unlike prior clustering approaches, the paper uses the backscattering side-channel, which has been shown to work better for hardware Trojan detection than other side-channels. We test the proposed algorithm on a set of 100 boards to thoroughly evaluate manufacturing variations among different hardware instances. This approach requires no priori knowledge about the chip or Trojan circuitry to cluster ICs into groups for HT detection. The results show that our technique can tolerate manufacturing variations among hardware instances to cluster all boards correctly for not only 9 different dormant Trojan designs on 3 different benchmark circuits from Trusthub, but also dormant Trojan designs whose trigger size is shrunk to as small as 0.19% of the original circuit.

CHAPTER 6

COUNTERFEIT IC DETECTION USING BACKSCATTERING SIDE-CHANNEL

6.1 Overview

Over the past few years, globalization of the semiconductor supply chain has led companies to outsource much of the production cycle for integrated circuits. While outsourcing helps companies significantly reduce their cost and time-to-market, it also introduces concerns about the trustworthiness of an ICs. One of the most serious problems is counterfeiting of ICs, which not only negatively impacts innovation and economic growth of the IC industry, but also creates serious threats and risks for systems that incorporate those counterfeit ICs.

This chapter proposes a novel method that uses the backscattering side-channel to cluster ICs such that counterfeits are separated from legitimate ICs. The backscattering side-channel, which has been introduced only recently, has been proven to outperform other side-channels in detecting hardware Trojan horses (HTs), i.e., ICs where additional logic gates (and connections to existing logic gates) have been added. In this work, we use it to robustly separate ICs into legitimate and counterfeit ones, even when only layout or placement of the IC has changed, without any added logic or connections. We evaluate our technique on a set of ten boards over six different counterfeit IC designs, and find that our technique tolerates manufacturing variations among different hardware instances, detecting counterfeit ICs with 100% accuracy and 0% false positives.

The rest of the chapter is organized as follows. Section 6.2 explains our new technique for counterfeit IC detection, Section 6.3 describes the setup for out experimental evaluation, Section 6.4 presents the results of that evaluation and, finally, Section 6.5 concludes the chapter.

6.2 A Novel Approach for Counterfeit Detection

6.2.1 Using Backscattering Side-Channel for Counterfeit IC Detection

Nguyen et al. [64] have shown that HTs can be detected by analyzing how impedance changes during a clock cycle, and comparing the impedance changes of the IC under test to those of a "golden" IC (i.e., an IC that is known to be free of HTs). As pointed out by Nguyen et al. [64], however, time-domain reception/recording of the backscattered signal for this would require extremely high bandwidth (many times the clock frequency of the IC), and the signal distortion due to radio-frequency (RF) noise, quantization noise (ADC resolution), imperfect synchronization (jitter), etc., would make it difficult to identify small changes caused by the presence of a stealthy HT. Thus, the backscattered signal was instead measured in the frequency domain, at multiple harmonics of the ICs clock frequency, which directly correspond (through Discrete Fourier Transform) to time-domain samples during the clock cycle. These frequency-domain measurements can be very accurate because they can be collected as separate frequency-bin measurements (with slower but more accurate ADCs), averaged over many clock cycles to reduce the impact of RF noise and jitter.

In these measurements, the change caused by an HT will be reflected in backscattered signals at the harmonics of the circuit's clock frequency: $f_{carrier} \pm f_c$, $f_{carrier} \pm 2 * f_c$, etc. The first clock harmonic at $f_{carrier} \pm f_c$ follows the overall RCS change during a cycle, while the remaining harmonics are affected by the rapidity of change (rise/fall times), and timing of the impedance changes within the clock cycle.

In the time domain, when state changes caused by HTs become briefer in duration, the corresponding frequency-domain changes at the clock harmonics become smaller in magnitude and shift to higher harmonics. Compared to lower harmonics, the higher harmonics tend to be affected more by noise, clock jitter, and other measurement impairments. One of the key reasons why the backscattering side-channel is highly suitable for HT detection (compared to traditional analog side-channels such as EM and power) is that impedance

changes, once they happen, persist for the rest of the cycle, so larger changes in impedance tend to affect relatively low harmonics of the clock in the backscattered signal. In contrast, the current bursts that create EM and power signals are already very brief, and the presence of an HT tends to change the magnitude of these bursts and/or shift their timing within the cycle, so the presence of the HT tends to only affect higher harmonics of the clock.

In this work, we follow the measurement approach of Nguyen et al. [64]. Specifically, for each circuit, we measure the amplitude of the first *N* harmonics of the clock from its backscattering side-channel signals to form a vector, which characterizes the circuit's overall amount, timing, and duration of impedance-change activity during a clock cycle. Thus, we can represent each circuit by a vector of *N* points, which are the amplitudes of the first *N* harmonics of the clock from its backscattering side-channel signals: $\mathbf{h} = [h_1, h_2, ..., h_{N-1}, h_N]$, where h_j is the amplitude of the j^{th} harmonic of the clock. These vectors will be used as inputs for our clustering algorithm that identifies counterfeit ICs, even when they only differ from legitimate ICs in layout or placement on the chip, without any additional gates (or connections among gates) compared to the original IC. This is in contrast to Nguyen et al. [64], which focused on detection of HTs, i.e., on detection of IC where additional logic gates (and connections to existing gates) are present.

6.2.2 One-Class-Classification to Detect Counterfeit ICs

In this section, we introduce our one-class-classification technique that accurately detects whether or not the IC's layout is the legitimate one. The approach is based on supervised learning techniques which contain two phases: training and testing. In the training phase, the goal is to obtain parameters of the cluster that correcponds to back-scattered signals at clock-frequency harmonics for a legitimate IC. The testing phase then determines whether measured back-scattered signals for an IC-under-test map within or outside of the legitimate-IC cluster.

To achieve our goal, we first collect magnitudes of the first N harmonics as described

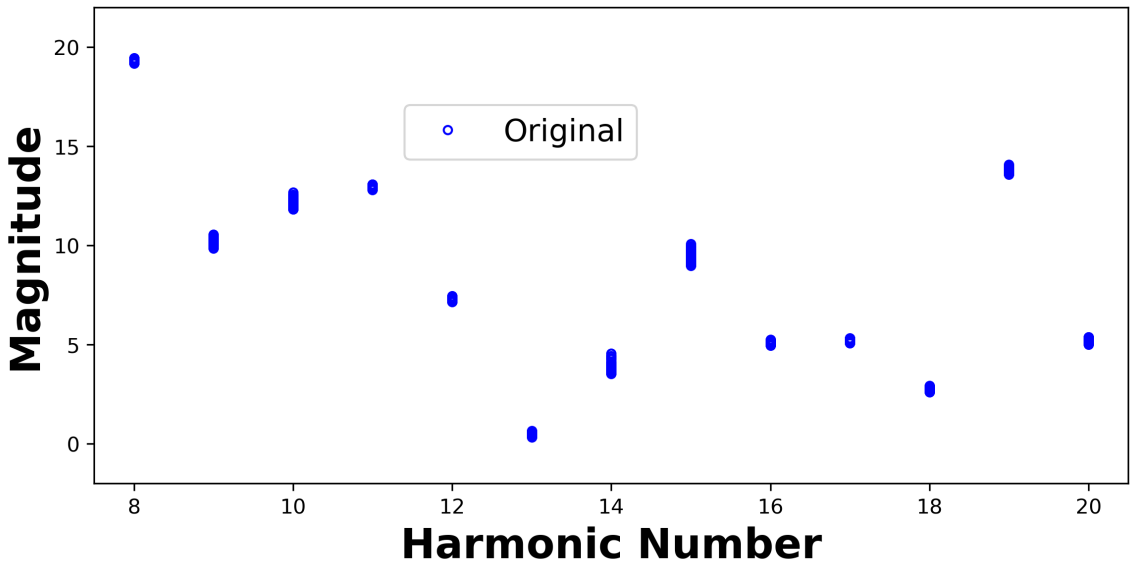


Figure 6.1: Harmonic magnitudes of the original circuit in the training phase.

in Section 6.2.1. An example of the original ICs harmonic magnitudes are given in Figure 6.1. We observe that the harmonics are generally dense around the mean for each harmonic yet magnitudes vary significantly among different harmonics. Therefore, our model first considers each harmonic independently, and then combines the results of harmonics to deduce the originality of the layout. In that respect, we calculate the mean magnitude of each harmonic as

$$\hat{\mathbf{h}}[k] = \frac{1}{M}\sum_{m=1}^{M}\mathbf{h}_m[k] \tag{6.1}$$

where M is the number of training measurements, and $\mathbf{h}_m$ is a row vector containing the harmonic values of the m^{th} measurement which can be written as

$$\mathbf{h}_m = \begin{bmatrix} h_m^0 & h_m^1 & \cdots & h_m^N \end{bmatrix} \tag{6.2}$$

and $\mathbf{h}_m[k] = h_m^k$. To proceed further, let assume $\mathbf{M}_D$ be the distance of the most deviated harmonic values from the mean among all training measurements such that

$$\mathbf{M}_D = \max\left\{\texttt{abs}\left(\mathbf{H} - \mathbf{1}\hat{\mathbf{h}}^T\right)\right\} \tag{6.3}$$

where $\max\{\bullet\}$ returns a row vector which contains the maximum values at each column of its argument, $\texttt{abs}\{\bullet\}$ returns the magnitude of its argument, $\mathbf{1} \in \Re^M$ is a column vector with full of ones, and $\mathbf{H}$ is a matrix such that each row represents a measurement, and each column contains the corresponding harmonic value.

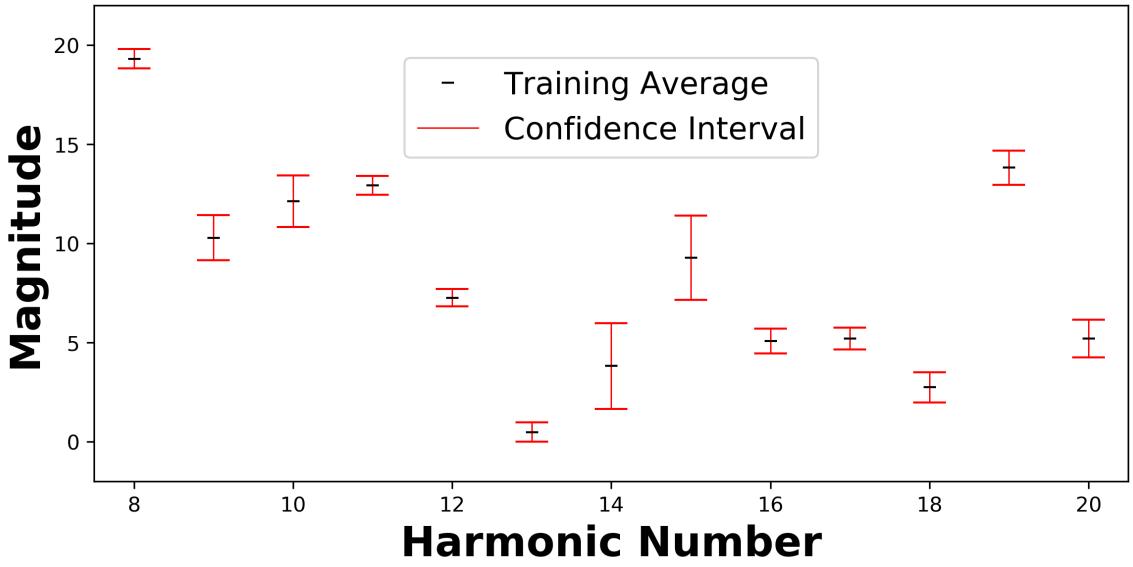


Figure 6.2: Average harmonic magnitudes and confidence intervals of the original circuit for each harmonic.

Density estimation, a common method in one-class-classification, works better with large numbers of measurements [115]. However, having such a large sample space is difficult and takes really long time to collect signals. Therefore, by mimicking the well-known "3σ" rule, we define the confidence interval for the m^{th} harmonic as

$$[\hat{\mathbf{h}}[m] - 3 \cdot \mathbf{M}_D[m], \quad \hat{\mathbf{h}}[m] + 3 \cdot \mathbf{M}_D[m]]. \tag{6.4}$$

An example of the intervals and the average harmonic signal is given in Figure 6.2. One of the main observations is that the spread around each harmonic varies, therefore, considering each harmonic independently gives better insight about the layout. For an even

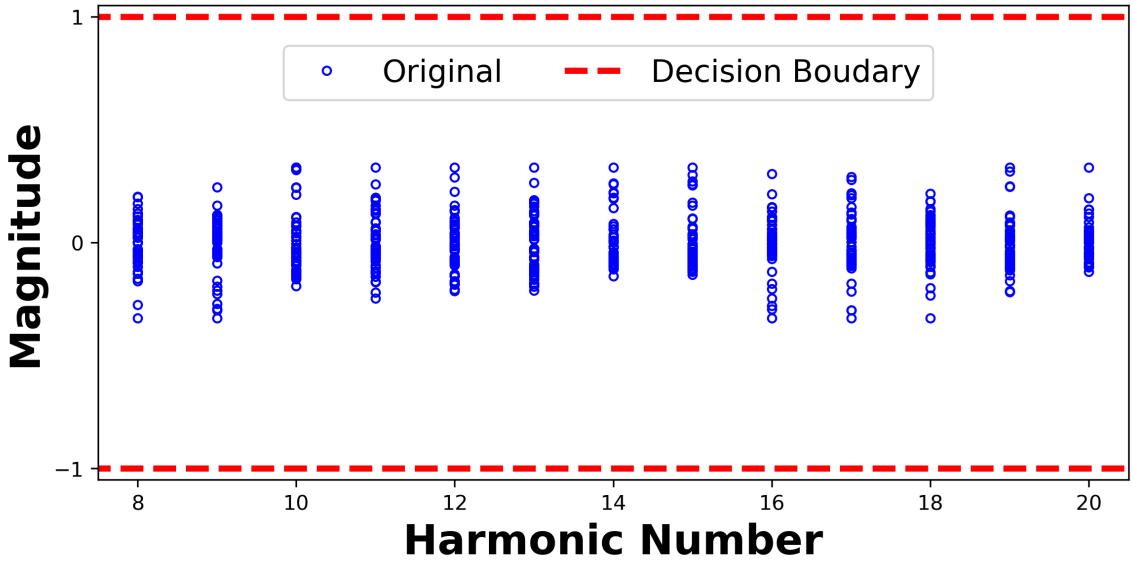


Figure 6.3: Normalized harmonic magnitudes and decision boundaries of the original circuit for each harmonic.

better illustration of the harmonics, we normalize the data as follows:

$$\mathbf{H} = (\mathbf{H} - \mathbf{1}\hat{\mathbf{h}}) \,./\, (\mathbf{1}\mathbf{M}_D) \tag{6.5}$$

where "./" is the pairwise division operation. After this normalization of the data, the confidence interval or decision boundaries are fixed to between -1 and 1. After the normalization, the training data and the boundaries are given in Figure 6.3. Finally, when testing an IC, the layout is called

- *Original* if all measured harmonics are within the corresponding confidence intervals,
- *Counterfeit* if there is at least one harmonic values which violates its confidence interval.

6.3 Benchmark Implementation and Experiment Setup

Counterfeit IC Benchmark Implementation

For our experimental evaluation, we implement two different types of counterfeit IC: 1) Counterfeit ICs with the same functionality as the original but different physical implementation (position) of the circuit, and 2) Counterfeit ICs with the same functionality and position as the original but different physical layout (routing and placement) of the circuit.

- Counterfeit ICs with Different Layout: We have implemented several counterfeit IC examples by re-compiling and letting the EDA tool to change the placement and routing of the circuit. We have four different test subject designs: Original layout AES IC, 1st layout AES counterfeit IC, 2nd layout AES counterfeit IC, 3rd layout AES counterfeit IC.

- Counterfeit ICs with Changed Position: We have implemented several counterfeit IC examples by moving the placement of the AES circuit from its original placement. We have four different test subject designs: original position AES IC, 1st position AES counterfeit IC, 2nd position AES counterfeit IC, and 3rd position AES counterfeit IC.

Experimental Setup

The experimental setup to evaluate the performance of the proposed algorithm is shown in Fig. 6.4. The setup includes a transmitter Aaronia E1 electric-field near-field probe [103] connected to an Agilent MXG N5183A signal generator [108], and a receiver Aaronia H2 magnetic field near-field probe [103] connected to an Agilent MXA N9020A spectrum analyzer [109]. The devices-under-test (DuT) are Altera DE0 Cyclone V FPGA boards [110]. An angle ruler is used as a positioner so that different DE0-CV boards can be tested using approximately the same position of probes. A laptop is used to control the devices

and automate the measurements. A 3 GHz continuous sinusoidal signal is generated by the signal generator, and backscattered signals are recorded by the spectrum analyzer.

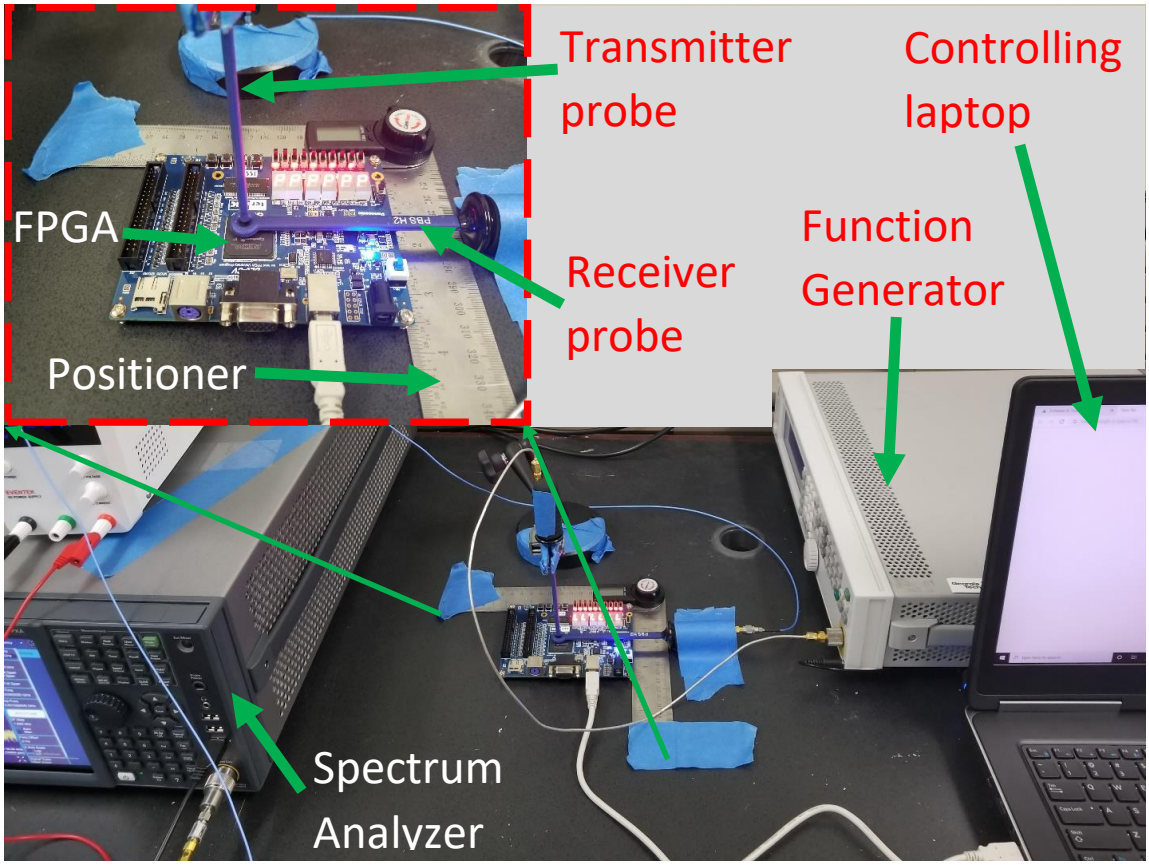


Figure 6.4: Measurement setup for counterfeit IC detection using backscattering side-channel.

6.4 Experimental Results and Discussion

In this section, we provide experimental results when layout or placement position of the circuit changes. We will first perform an experiment when the layout of the circuit changes while keeping functionality the same. The results are given in Figure 6.5-6.8. Recall that the harmonic values and the boundaries are normalized based on the equation given in (6.5). During testing, in this equation $\mathbf{H}$ is a matrix, where each row is a test measurement, $\hat{\mathbf{h}}$ is the mean harmonic values obtained from training measurements, and $\mathbf{M}_D$ is the maximum deviation row vector for each harmonic which is also obtained in the training phase. Figure

6.5 shows the harmonics for the original layout, along with confidence intervals obtained from training. We observe that all considered harmonics are well within the confidence interval, and thus all these measurements are labeled as corresponding to the original IC (**0%** false positive rate).

Next, we experiment with counterfeit layouts, which have the same functionality as the original IC but different layout. Figures 6.6, 6.7 and 6.8 illustrate the harmonic values for different counterfeits of this kind. We observe that all of the measurements have at least one component that violates the confidence intervals (**100%** true positives, i.e. **100%** accuracy in detecting counterfeits of this kind).

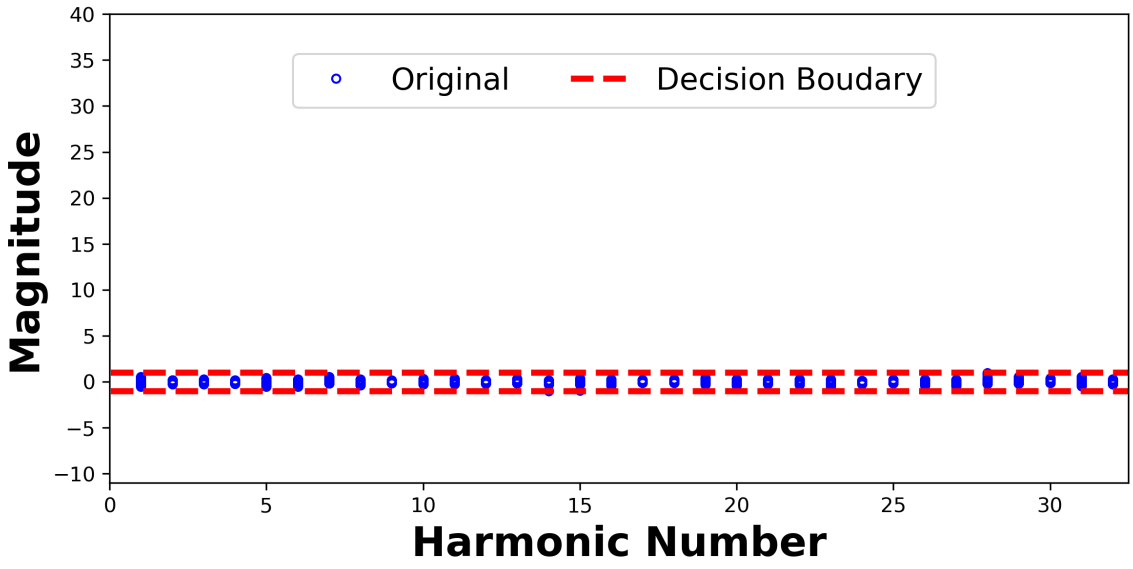


Figure 6.5: Normalized harmonic magnitudes of the original circuit.

Another experiment is performed by changing the placement location of the circuit within the chip, while keeping its functionality and layout same. The results are given in Figure 6.9-6.12. Figure 6.9 shows the results of our testing for instances of the original circuit which, like in Figure 6.6, correctly labels all these instances as original. Figure 6.10-6.12 show the results of position-change counterfeits. For each of these counterfeits, at least one harmonic is outside the confidence interval, causing all these counterfeits to be accurately labeled as counterfeit. Overall, for this type of counterfeit IC our method again

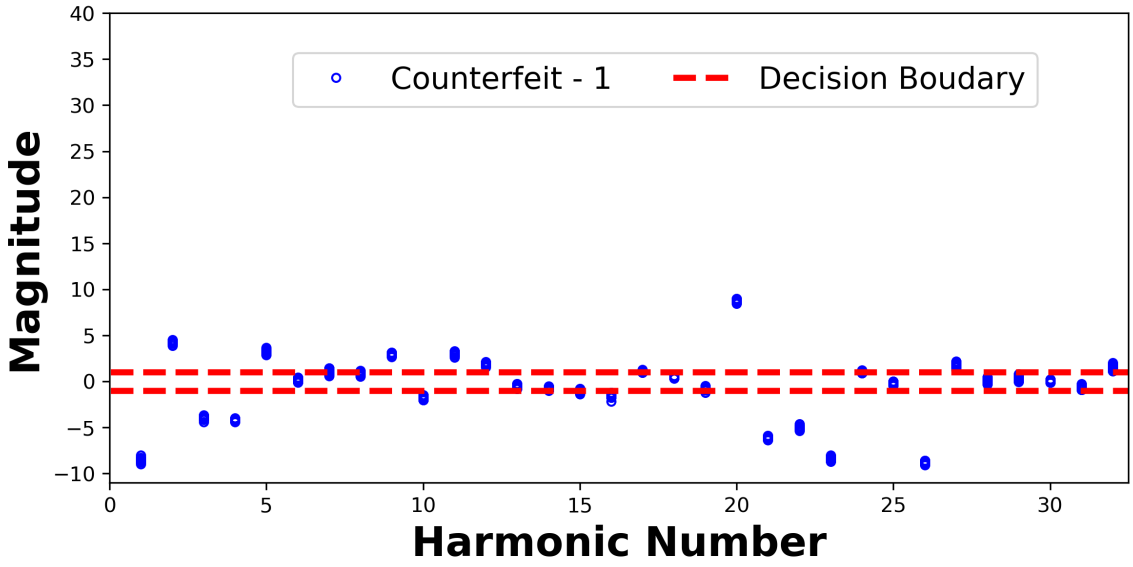


Figure 6.6: Normalized harmonic magnitudes of the circuit with the same functionality and a different layout.

achieves **100%** detection accuracy.

In summary, both experiments reveal that our methodology is a very powerful and robust to identify the counterfeit circuits.

6.5 Conclusions

Over the past few years, globalization of the semiconductor supply chain has led companies to outsource much of the production cycle for integrated circuits (ICs). While outsourcing helps companies significantly reduce their cost and time-to-market, it also introduces concerns about the trustworthiness of an IC. One of the most serious problems is counterfeiting of ICs, which not only negatively impacts innovation and economic growth of the IC industry, but also creates serious threats and risks for systems that incorporate those counterfeit ICs. This chapter proposes a novel method that uses the backscattering side-channel to cluster ICs such that counterfeits are separated from legitimate ICs. The backscattering side-channel, which has been introduced only recently, has been proven to outperform other side-channels in detecting hardware Trojan horses (HTs), i.e., ICs where additional

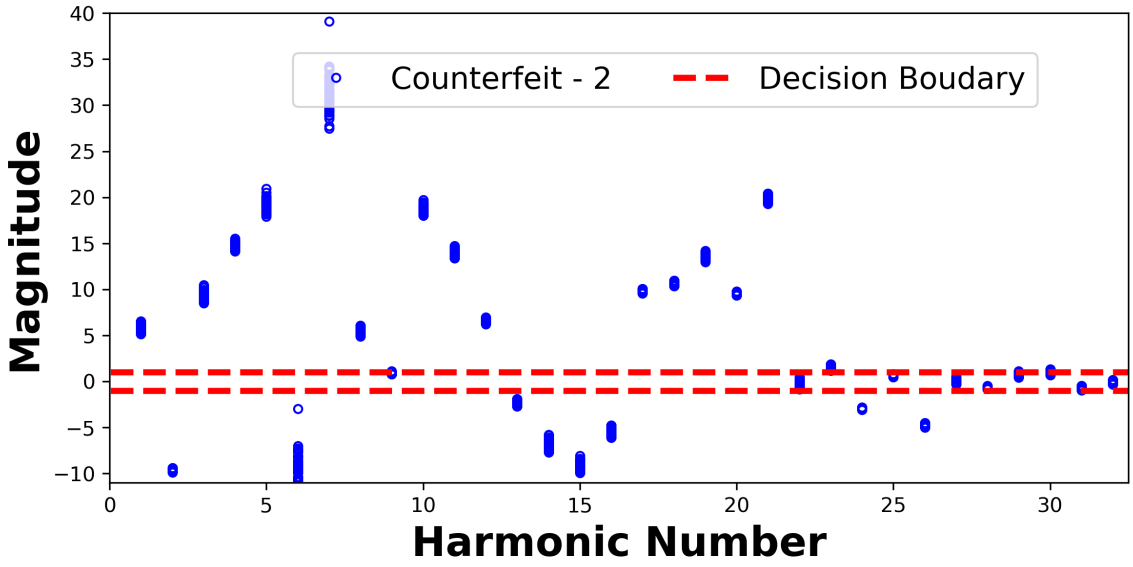


Figure 6.7: Normalized harmonic magnitudes of the circuit with the same functionality and a different layout.

logic gates (and connections to existing logic gates) have been added. In this work we use it to robustly separate ICs into legitimate and counterfeit ones, even when only layout or placement of the IC has changed, without any added logic or connections. We evalute our technique on a set of ten boards over six different counterfeit IC designs, and find that our technique tolerates manufacturing variations among different hardware instances, detecting counterfeit ICs with 100% accuracy and 0% false positives.

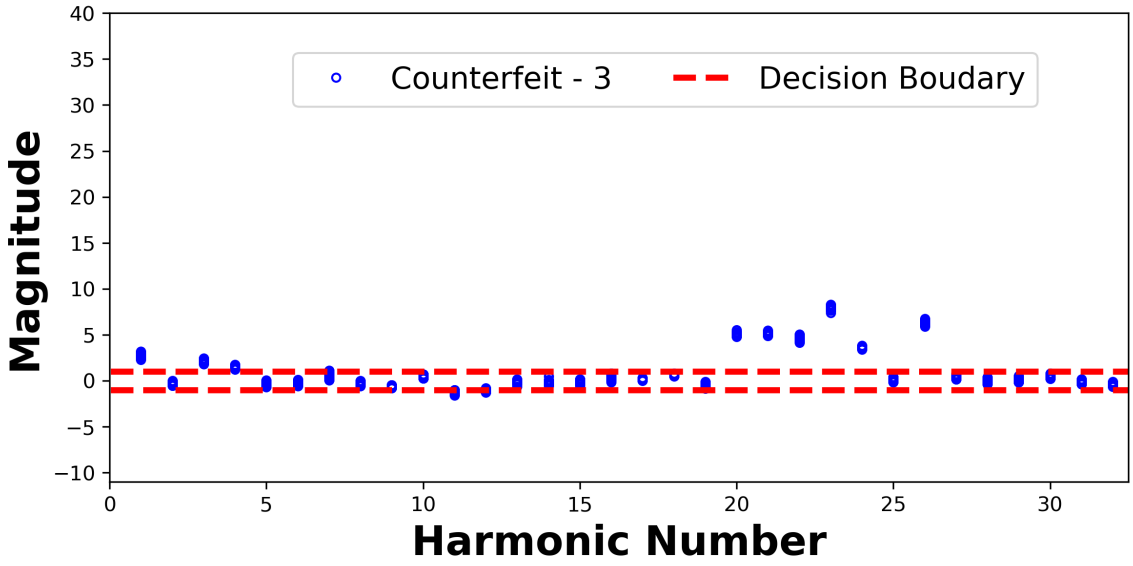


Figure 6.8: Normalized harmonic magnitudes of the circuit with the same functionality and a different layout.

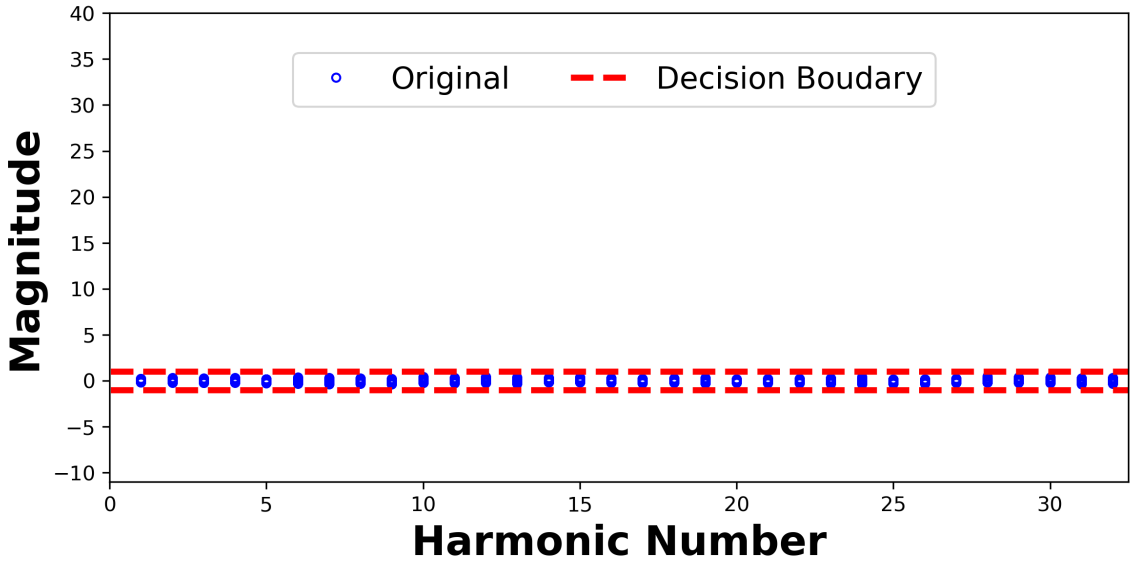


Figure 6.9: Normalized harmonic magnitudes of the original circuit.

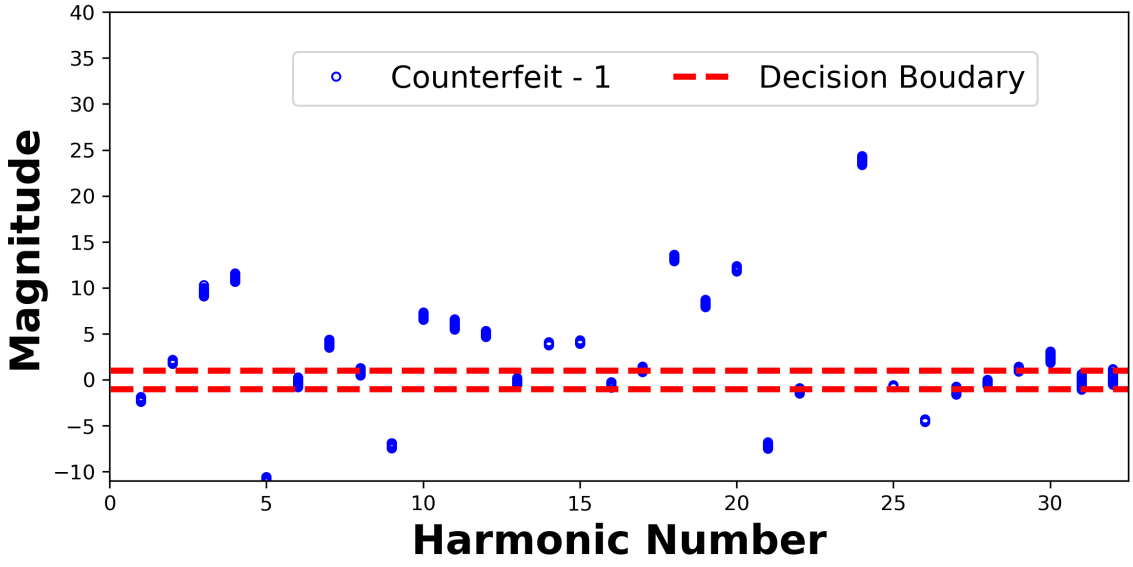


Figure 6.10: Normalized harmonic magnitudes of the counterfeit circuit with the same functionality and and layout, but different placement position 1.

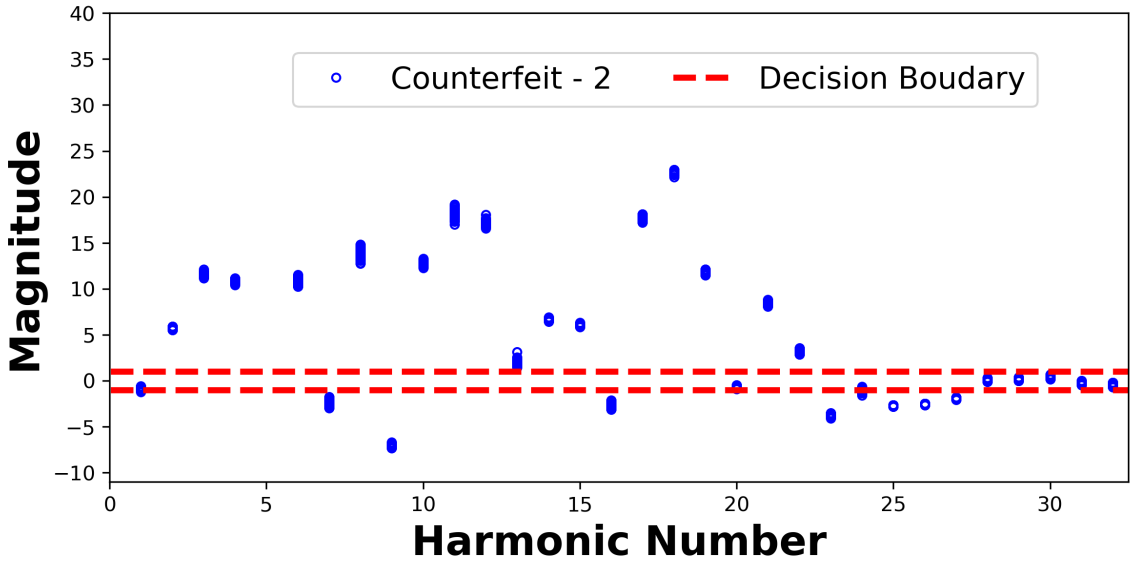


Figure 6.11: Normalized harmonic magnitudes of the counterfeit circuit with the same functionality and and layout, but different placement position 2.

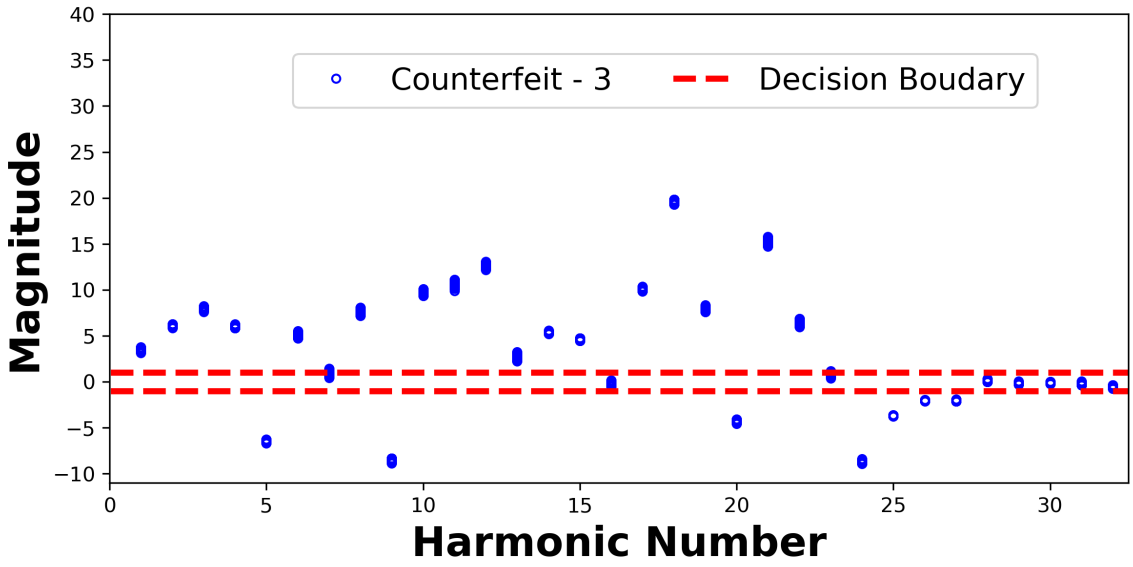


Figure 6.12: Normalized harmonic magnitudes of the counterfeit circuit with the same functionality and and layout, but different placement position 3.

CHAPTER 7
GOLDEN-CHIP-FREE HARDWARE TROJAN DETECTION TECHNIQUE USING BACKSCATTERING SIDE-CHANNEL

7.1 Overview

Among hardware Trojan detection techniques, side-channel analysis approaches are the most widely used ones because they are non-destructive, relatively cheap, and fast, which is suitable for testing a large number of ICs. Especially, externally-measured side-channel analysis approaches are dominantly preferred because they require no modifications to the chip circuitry. Externally-measured side-channel analysis methods rely on measuring some non-functional properties from outside the IC as it operates, and comparing the measurements to reference signals produced by either simulation or a verified genuine device. Because it is exceedingly difficult to model the EM and power signal of digital circuit, most of the existing side-channel analysis based HT detection techniques rely on the assumption of having a golden (HT-free) chip for training. This assumption is too strong, and often unrealistic, which prevents them from being used for practical deployments of HT detection.

Motivated by the shortcomings of previous techniques, this chapter proposes a novel golden-chip-free hardware Trojan detection technique using backscattering side-channel with circuit impedance modeling. We use the backscattering side-channel because the backscattering side-channel outperforms other side-channels in hardware Trojan detection, as demonstrated in the previous chapters. In addition, the backscattering side-channel is impedance-based, while other traditional side-channels such as EM and power are current-flow based. It is relatively easier to model and predict the impedance changes than the current-flow changes of a circuit. Furthermore, when a HT is attached to the circuit, it

changes the circuit impedance, regardless of whether it is activated or not. That is why we decided to use the backscattering side-channel for our golden-chip-free hardware Trojan detection technique.

Unlike previous techniques, we build models that help calculate the reference impedances of benchmark circuits. It is much less complicated and more accurate to estimate the impedance of a circuit than estimating the current-flow changes in a complex IC. Using the models, we estimate the impedances of benchmark circuits, and then estimate the expected powers of the backscattering side-channel signal of clock harmonics. Then we compare the measurements against the models to detect HTs without having to have a golden sample.

We start with simple circuits such as transistor and inverter, then build up to impedance models for more complex circuits. These models are used to calculate the reference impedances of the circuits, and then the expected values of the backscattered signal power of the clock harmonics. The models are compared against the measurements. Our algorithms report a design as Trojan-free if the measurements match the model, and report a design as Trojan-infected otherwise.

The rest of this chapter is organized as follows. Section 7.2 discusses the first order analysis of digital circuits and theoretical assumptions for our impedance model. Section 7.3 presents our impedance model, while Section 7.4 introduces the golden-chip-free HT detection techniques. Sections 7.5 and 7.6 evaluate the effectiveness of the technique on different Trojan benchmarks. Finally, Section 7.7 concludes this chapter.

7.2 First Order Analysis of Digital Circuits

Integrated circuits consist of millions or even billions transistors, and even a single deep-submicron transistor itself is considered as a complex device. Its behavior is heavily non-linear and is influenced by a large number of second-order effects [116]. Therefore, the question that needs to be answered is how to abstract the behavior of MOS transistors, and thus digital circuits, into a simple and tangible analytical model that does not lead to

exceedingly complicated equations, yet captures the essentials of the circuit. It turns out that the first-order expressions can be combined into a single expression that meets these goals [117].

Analyzing the behavior of integrated circuits consisting of billions transistors is difficult and cumbersome. Therefore, the following substantial simplifications are made in order to enable modeling of integrated circuits without intolerable accuracy degrades:

- A transistor can be considered as a switch controlled by its gate signal. An NMOS transistor is on when the gate signal is high and is off when the gate signal is low. Inversely, a PMOS transistor is on when the gate signal is low and off when the gate signal is high as illustrated in Fig 7.1 [117].

- The equivalent effective resistance is the ratio of V_{ds} to I_{ds} averaged across the switching interval of interest. Therefore, the transistor is nothing more than a switch with an infinite equivalent effective off-resistance (for $\mid V_{gs} \mid < \mid V_T \mid$), and a finite equivalent effective on-resistance (for $\mid V_{gs} \mid > \mid V_T \mid$) [117].

- The logic levels do not depend on the relative device sizes, so that the transistors can be minimum size. As a result, all transistors in the circuit have the same size, hence the same equivalent effective resistance [116].

- Long-channel model is used and all short-channel effects can be ignored. As a result, the resistance of two transistors in series is the sum of the resistances of each transistor, and the total resistance of a set of resistors in parallel is found by adding up the reciprocals of the resistance values, and then taking the reciprocal of the total [116].

- No direct path exists between the supply and ground rails under steady-state operating conditions (this is, when the input and outputs remain constant). The absence of current flow (ignoring leakage currents) means that the gate does not consume any static power [117].

- Interconnect assumptions [117]:
  - Inductive effects can be ignored if the resistance of the wire is substantial. This is for instance the case for long Aluminum wires with a small cross-section, or if the rise and fall times of the applied signals are slow.
  - When the wires are short, the cross-section of the wire is large, or the interconnect material used has a low resistivity, a capacitance-only model can be used.
  - when the separation between neighboring wires is large, or when the wires only run together for a short distance, inter-wire capacitance can be ignored, and all the parasitic capacitance can be modeled as capacitance to ground.
  - No long interconnect is present in the circuit.

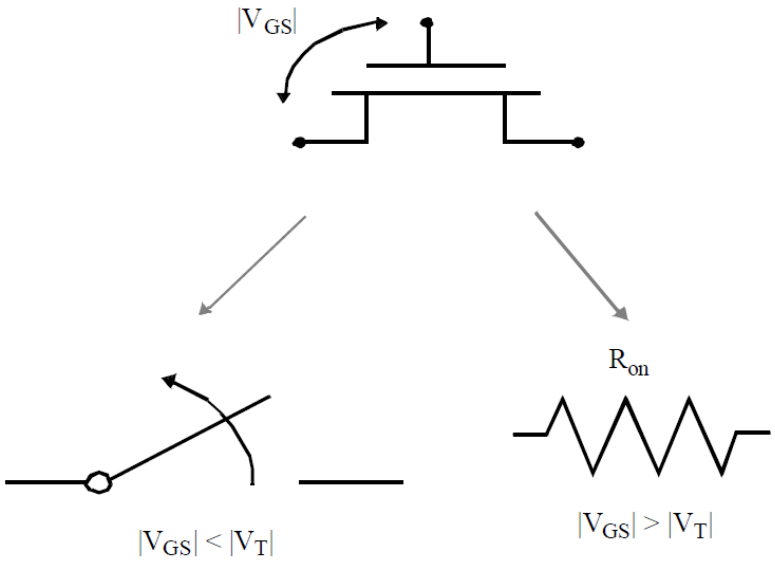


Figure 7.1: Switch Model of CMOS Transistor [117].

Based on these assumptions, it is possible to do first order analysis to build models for estimating the impedance, which includes equivalent resistance and capacitance, of a circuit.

7.2.1 Equivalent Effective Resistance

As mentioned above, our model is based on the assumption that the transistor is nothing more than a switch with an infinite equivalent effective off-resistance, and a finite equivalent effective on-resistance R_{eq-on}. Because the on-resistance of the transistor is still time-variant, non-linear and depending upon the operation point of the transistor, we define R_{eq-on} as an average value of the resistance over the operation region of interest. If the resistance does not experience any strong non-linearities over the range of the averaging interval, the equivalent effective resistance R_{eq-on} can be calculated by the average value of the resistances at the end-points of the transition.

$$R_{eq-on} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} R_{on}(t)dt = \frac{1}{t_2 - t_1} \int_{t_2}^{t_1} \frac{V_{DS}(t)}{I_D(t)}dt, \tag{7.1}$$

Where V_{DS} is the voltage between the drain and the source of transistor, I_D is the current flowing through the channel when the transistor is on. We are interested in the point where the voltage on the capacitor reaches the mid-point $(V_{DD}/2)$, where V_{DD} is the supply voltage. We assume that the the saturation voltage V_{DSAT} of the transistor and the transistor stays in saturation for the entire duration of the transition. Then equation 7.1 can be expressed as follows.

$$R_{eq-on} = \frac{1}{-V_{DD}/2} \int_{V_{DD}}^{V_{DD}/2} \frac{V}{I_{DSAT}(1 + \lambda V)}dV \approx \frac{3}{4}\frac{V_{DD}}{I_{DSAT}}(1 - \frac{7}{9}\lambda V_{DD}), \tag{7.2}$$

$$I_{DSAT} = k'\frac{W}{L}((V_{DD} - V_T)V_{DSAT} - \frac{V_{DSAT}^2}{2}, \tag{7.3}$$

$$k' = \mu C_{ox} = \mu\frac{\epsilon_{ox}}{t_{ox}}, \tag{7.4}$$

where I_{DSAT} is the saturated current, k' is the the process transconductance parameter, W is the width of the transistor, L is the channel length of the transistor, C_{ox} stands for the capacitance per unit area presented by the gate oxide, $\epsilon_{ox} = 3.97\epsilon_o = 3.5 * 10^{11} F/m$ is the oxide permittivity, t_{ox} is the thickness of the oxide, μ is the mobility of the channel of an transistor, V_T is the threshold voltage of the transistor, and λ is an empirical parameter, called the channel-length modulation. The product of the process transconductance and the (W/L) ratio of an transistor is called the gain factor of the device. λ varies roughly with the inverse of the channel length. In other words, the drain-junction depletion region presents a larger fraction of the channel in shorter transistors, thus the channel-modulation effect is more pronounced. λ can be considered as zero because we assume that all short-channel effects can be dismissed. From the formula, we can see that the resistance is inversely proportional to the (W/L) ratio of the device. For a specific technology (45 nm, 22 nm, etc.), the channel length L is fixed, only the width of the transistor can be changed. As a result, doubling the transistor width halves the resistance.

Table 7.1 summarizes the parameters needed for estimating the resistance of a transistor in 22 nm technology. As discussed, all the parameters except the transistor width are fixed for a specific technology. Table 7.2 shows resistance of a transistor in 22 nm technology with different values of transistor width. In the table, R_N is the equivalent on-resistance of an NMOS transistor, and R_P is the equivalent on-resistance of an PMOS transistor. As we can see, with the same transistor width, the on-resistance of PMOS is bigger than NMOS because PMOS has lower mobility compared to NMOS. As a result, normally PMOS is sized to a bigger size so the values of R_N and R_P are similar, thus making the rise-time deday and fall-time delay similar.

7.2.2 Equivalent Capacitance

It is nearly impossible to manually analyze MOS circuits where each capacitor is considered individually. Therefore, we assume that all capacitances can be lumped together into

Table 7.1: Parameters needed for transistors' resistance estimation

Parameter	NMOS	PMOS
L	22	22
μ	$2.5e^6$	$2.5e^6$
ϵ_{ox}	3.9	3.9
t_{ox}	$1.08e^{-9}$	$1.1e^{-9}$
V_{DSAT}	0.17	0.17
V_{DD}	1	1
V_T	0.68858	-0.63745

Table 7.2: Estimated resistances for different values of W/L

Resistances	**Values**			
	$(W/L) = 2$	$(W/L) = 5$	$(W/L) = 10$	$(W/L) = 20$
R_P	17.41 $k\Omega$	6.964 $k\Omega$	3.482 $k\Omega$	1.741 $k\Omega$
R_N	7.213 $k\Omega$	2.885 $k\Omega$	1.442 $k\Omega$	0.721 $k\Omega$

one single capacitor C_L, located between V_{out} and GND.

Figure 7.2 shows all the capacitances of transistors in a NOT gate, in which C_{gd12} is the gate-drain capacitance, C_{db1} and C_{db2} are the diffusion capacitances between drain and bulk, C_w is the capacitance due to the wiring, and C_{g3}, C_{g4} are the gate capacitances of fanout. Our model assumes that all capacitances are lumped together into one single capacitor C_L, located between V_{out} and GND as illustrated in Fig. 7.3. This load capacitance C_L is the combination of C_{gd12}, C_{db1}, C_{db2}, C_w, C_{g3}, and C_{g4}.

Gate-Drain Capacitance

Under the assumption that transistors are either in cut-off or in the saturation mode during switching transition, the only contributions to C_{gd12} are the overlap capacitances because the channel capacitance of the MOS transistors does not play a role here, as it is located either completely between gate and bulk (cut-off) or gate and source (saturation) [117]. Therefore, the gate-drain capacitance can be calculated as illustrated in equation 7.5, where

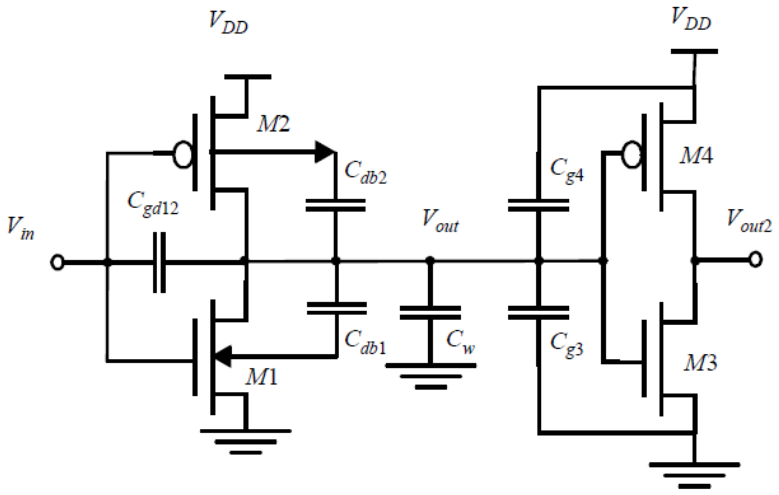


Figure 7.2: Parasitic capacitances of transistors in a NOT gate [117].

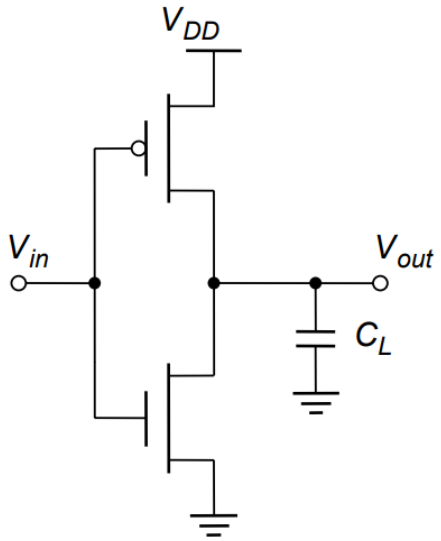


Figure 7.3: Equivalent lumped capacitance.

$CGDO_n$ and $CGDO_p$ are the gate-drain overlap capacitance per unit area.

$$C_{gdn} = CGDO_n W_n, C_{gdp} = CGDO_p W_p \tag{7.5}$$

Considering the Miller effect, this floating gate-drain capacitor can be replaced by a capacitance-to-ground capacitor whose capacitance is twice the actual gate-drain capacitance. This is because during a low-high or high-low transition, the terminals of the gate-drain capacitor are moving in opposite directions as illustrated in Fig. 7.4. As a result, the

equivalent capacitance to ground can be calculated as shown in the equation 7.6.

$$C_{Mgdn} = 2 * CGDO_n W_n, C_{Mgdp} = 2 * CGDO_p W_p \tag{7.6}$$

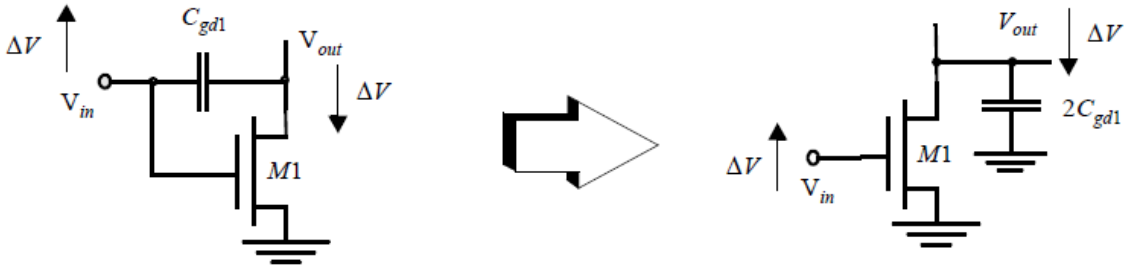


Figure 7.4: The Miller effect: Equivalent capacitance-to-ground capacitor of the gate-drain capacitor [117].

Diffusion Capacitances

Due to the reverse-biased pn-junction between the drain and the body (bulk) of the transistor, there is a diffusion capacitance between the drain and the body. This capacitance can be replaced by a linear one with the same change in charge for the voltage range of interest [117]. Assuming that C_{j0}, C_{jsw0} are the bottom junction capacitance per unit area and sidewall junction capacitance per unit area under zero-bias conditions, AD is the drain area, and PD is the perimeter of the drain area. Equation 7.7 shows the relationship between the linearized capacitor and the value of the junction capacitance under zero-bias conditions,i.e.,

$$C_{eq} = K_{eq} AD * C_{j0} + K_{eqsw} PD * C_{jsw0}, \tag{7.7}$$

where K_{eq} and K_{eqsw} are multiplication factors which depend on the junction potential and the grading coefficient of the junction.

Wiring Capacitance

The capacitance due to wiring depends upon the length and width of the connecting wires, and is a function of the distance of the fanout from the driving gate and the number of fanout gates and is normally obtained by extraction from the design. Without knowing the routing, placement and the design itself, it is impossible to estimate the value of wiring capacitance. However, we can simplify the problem by assuming that there is no long interconnect in the design. As a result the wiring capacitance is small compared to other capacitances. The wiring capacitance can be assumed to be half the value of the gate-drain capacitance of an NMOS transistor [117, 118].

Fanout Capacitances

By assuming that all components of the gate capacitance are connected between $Vout$ and GND and the channel capacitance of the connecting gate is constant over the interval of interest, the fanout capacitance equals the total gate capacitance of the loading gates as illustrated in the following equation:

$$C_{fanout} = \sum_i C_{Gi} = \sum_i (C_{GSOi} + C_{GDOi} + W_i L C_{ox}) = \sum_i (CGSO_i + CGDO_i + LC_{ox}) W_i, \tag{7.8}$$

where C_{Gi} is the gate capacitance of the loading transistor i that is driven by the transistor whose fanout capacitance needs calculated. C_{Gi} includes gate-source overlap capacitance C_{GSOi}, gate-drain overlap capacitance C_{GDOi}, and gate-channel capacitance $W_i L C_{ox}$. $CGSO_i$ and $CGDO_i$ are the gate-source and gate-drain overlap capacitance, respectively.

Table 7.3 summarizes the capacitances that contribute to the total C_L and their formula. Note that when a gate switches from high to low, C_L has a different value when the inverter switches from low to high. The total impedance of a circuit is the combination of resistance and capacitance is calculated as $Z = R + 1/(j\omega C) = R + 1/(j2\pi f C)$, where f is the frequency. Table 7.2 shows two different impedance state of the circuit in 22 nm technology

Table 7.3: Capacitances that contributes to the total C_L

Capacitor	Expression
C_{Mgd1}	$2 * CGDO_n W_n$
C_{Mgd2}	$2 * CGDO_p W_p$
C_{db1}	$K_{eqn} AD_n C_{jn} + K_{eqswn} PD_n C_{jswn}$
C_{db2}	$K_{eqp} AD_p C_{jp} + K_{eqswp} PD_p C_{jswp}$
C_{fanout}	$\sum_i (CGSO_i + CGDO_i + LC_{ox}) W_i$
C_w	From Extraction
C_L	$\sum$

Table 7.4: Estimated impedances for different values of W/L

Impedance	**Value**			
	$(W/L) = 2$	$(W/L) = 5$	$(W/L) = 10$	$(W/L) = 20$
R_1	17.41 $k\Omega$	6.964 $k\Omega$	3.482 $k\Omega$	1.741 $k\Omega$
R_0	7.213 $k\Omega$	2.885 $k\Omega$	1.442 $k\Omega$	0.721 $k\Omega$
C_{L1}	0.182 fF	0.383 fF	0.647 fF	1.175 fF
C_{L0}	0.184 fF	0.386 fF	0.652 fF	1.184 fF
$1/(j\omega C_{L1})$	-j290.724 $k\Omega$	-j138.44 $k\Omega$	-j81.974 $k\Omega$	-j45.146 $k\Omega$
$1/(j\omega C_{L0})$	-j288.498 $k\Omega$	-j137.38 $k\Omega$	-j81.336 $k\Omega$	-j44.791 $k\Omega$
$Z_1 = R_1 + \frac{1}{j\omega C_{L1}}$	17.41-j290.724 $k\Omega$	6.964-j138.44 $k\Omega$	3.482-j81.974 $k\Omega$	1.741-j45.146 $k\Omega$
$Z_0 = R_0 + \frac{1}{j\omega C_{L0}}$	7.213-j288.498 $k\Omega$	2.885-j137.38 $k\Omega$	1.442-j81.336 $k\Omega$	0.721-j44.791 $k\Omega$

with different values of transistor width and frequency of $3.051\ Ghz$.

7.3 Impedance Model

Digital circuits are built by transistors, therefore, we use a transistor as the unit to build the impedance model of digital circuits. Cheng et al. [119] introduce a modulation loss factor, M, which relates the backscattered signal power to transistors' impedance variation of the circuit. M can be expressed as

$$M = \frac{1}{4} \left| \frac{Z_1(x) - 377}{Z_1(x) + 377} - \frac{Z_0(x) - 377}{Z_0(x) + 377} \right|^2, \tag{7.9}$$

where $Z_1(x)$ and $Z_0(x)$ are the estimated impedances of switching states of the circuit implemented on FPGA. An FPGA chip consists of logic blocks, which are arranged in a two dimensional grid and are connected by a programmable routing interconnect. This symmetrical grid is connected to I/O blocks which make off-chip connections. The "programmable/reconfigurable" term in FPGAs indicates their ability to implement a new function on the chip after its fabrication is complete. From the power source point of view, all these blocks are in parallel. Therefore, let parameter x represent the percentage of total logic resources being configured. $Z_1(x)$ and $Z_0(x)$ are defined as the following equations [119], i.e.,

$$Z_1(x) = \frac{Z_1}{x} + Z_{pkg}, \tag{7.10}$$

$$Z_0(x) = \frac{Z_0}{x} + Z_{pkg}, \tag{7.11}$$

where Z_1 and Z_0 are the estimated impedance of switching state of a circuit whose size is one percentage of total resources are utilized. On the other hand, the impedances Z_1 and Z_0 can be expressed as

$$Z_1 = R_1 + \frac{1}{j\omega C_1}, Z_0 = R_0 + \frac{1}{j\omega C_0}, \tag{7.12}$$

where R_1 and R_0 are switching resistances, C_1 and C_0 are switching capacitances, respectively. Z_{pkg} is the estimated resistance contributed by the package of the IC chip, e.g., wire bonds inside the chip case. The input impedance of the tag is equal to free space impedance, 377 Ω. In order to estimate the power of the backscattered signal from M, we introduce a coefficient α that relates modulation factor M to the power P.

$$P_f = \alpha(r, f, P_c) * M, \tag{7.13}$$

where P_f is the power of the backscattered signal, $\alpha(r, f, P_c)$ is the coefficient that depends on the distance of the probes to the DuT r, the frequency f, and the carrier power P_c. For a specific frequency, since we keep the probe positions and the carrier power constant, α is a constant as well.

To estimate those constants and build the impedance model, we implement a simple circuit consisting of a flip-flop and an inverter as illustrated in Fig. 7.5. This simple circuit switches between two impedance states, i.e., state 1 when the flip-flop is high and the inverter is low, and state 2 when the flip-flop is low and the inverter is high. In order to increase the logic utilization (the size of the circuit), which helps increase the strength of the backscattered signal caused by switching activities, we combine multiple pairs of flip-flops and inverters together, as illustrated in Fig. 7.6.

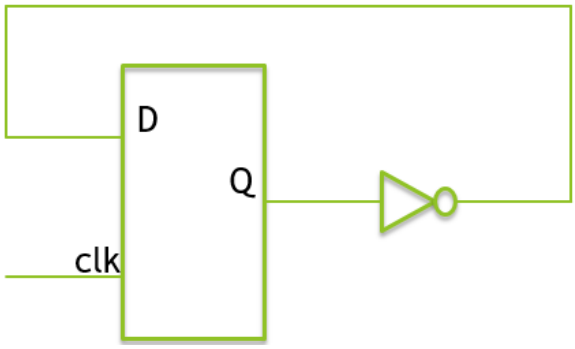


Figure 7.5: Simple flip-flop and inverter pair circuit.

We change the number of pairs of flip-flops and inverters to change the size of the circuit, and thus changing the signal strength caused by switching activities. As a result, we have multiple designs with different sizes and logic utilizations. These designs are used to program an FPGA and then we transmit a continuous wave at a frequency 3.031 GHz toward the FPGA. We then measure the corresponding backscattered power at multiple clock harmonics and peaks between clock harmonics. We then perform curve fitting to estimate the values of $R_1, R_0, \frac{1}{j\omega C_1}, \frac{1}{j\omega C_0}$ [119]. Fig. 7.7 shows the fitting curve after performing curve-fitting for the measurements. As we can see, the fitting curve estimates well the real measurement data. The estimated results of $R_1, R_0, \frac{1}{j\omega C_1}, \frac{1}{j\omega C_0}$ are (7.024 $k\Omega$, 3.364 $k\Omega$,

100.2 $k\Omega$, 98,67 $k\Omega$) which is within the range estimated in Section 7.2 and in [117].

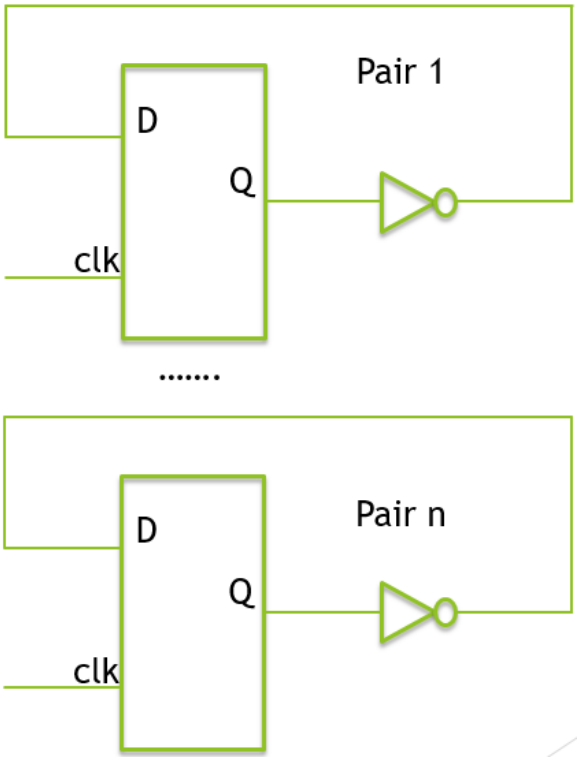


Figure 7.6: Combination of multiple flip-flop and inverter pair circuit.

7.4 Golden-chip-Free Hardware Trojan Detection Technique

Our golden-chip-free hardware Trojan detection technique includes two phases: estimation and detection.

7.4.1 Estimation

In the estimation phase, our technique uses the model in Section 7.3 to estimate the impedances from the input circuit. The impedances are estimated based on the logic utilization on FPGA of the circuit by using the equation 7.10. The parameters in the equation are estimated by performing cur-fitting as illustrated in Section 7.3. Then, the model estimates the power of the backscattered signal of m clock harmonics and the confidence interval (normally $\pm$ 5% of the power). Finally, it outputs the power and the confidence interval

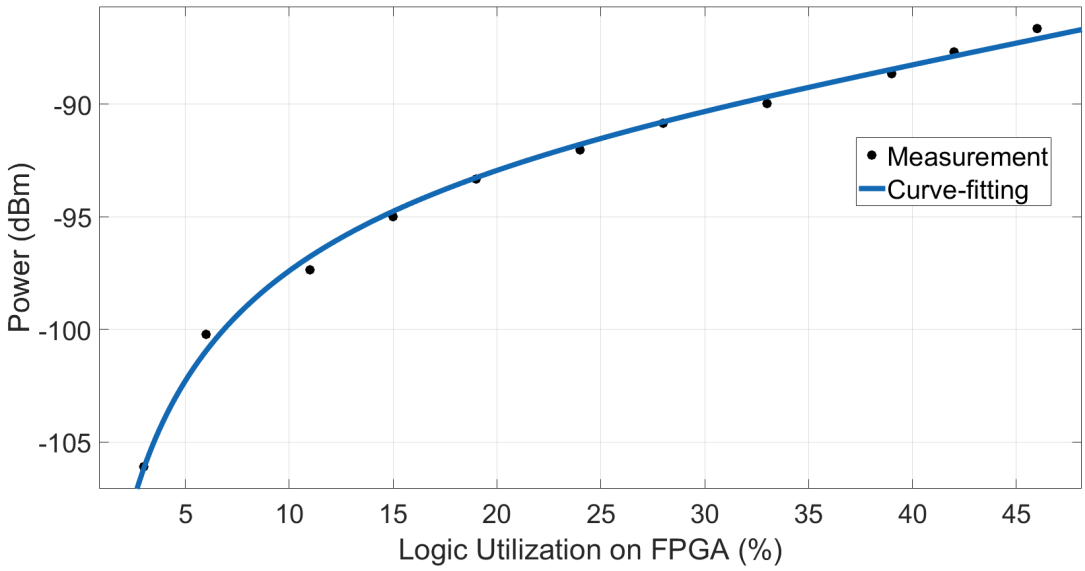


Figure 7.7: The fitting curve.

for each clock harmonics. These outputs will be the inputs of the detection phase. The summary of this estimation phase is illustrated in Fig. 7.8.

7.4.2 Detection

First, a single measurement is obtained of the m amplitudes that correspond to the lowest m harmonics of the IC's clock frequency in the side-band of the signal that is backscattered from the IC under test. Next, for each of the m amplitudes, we compute how much it deviates from the corresponding estimated value computed during the estimation phase. This deviation is computed as the absolute value of the difference, and intuitively it measures how much that amplitude differs from what would be expected from an HT-free IC. Finally, the sum of these deviations is compared to the sum of the confidence interval from estimation phase. The sum of the differences for the IC under test is a measure of how much its overall backscattering "signature" differs from what would be expected from an HT-free IC, and the sum of confidence interval from estimation phase corresponds to how much an individual measurement of an HT-free IC can be expected to differ from the average of HT-free measurements. The IC under test is labeled as HT-free if its sum of deviations is

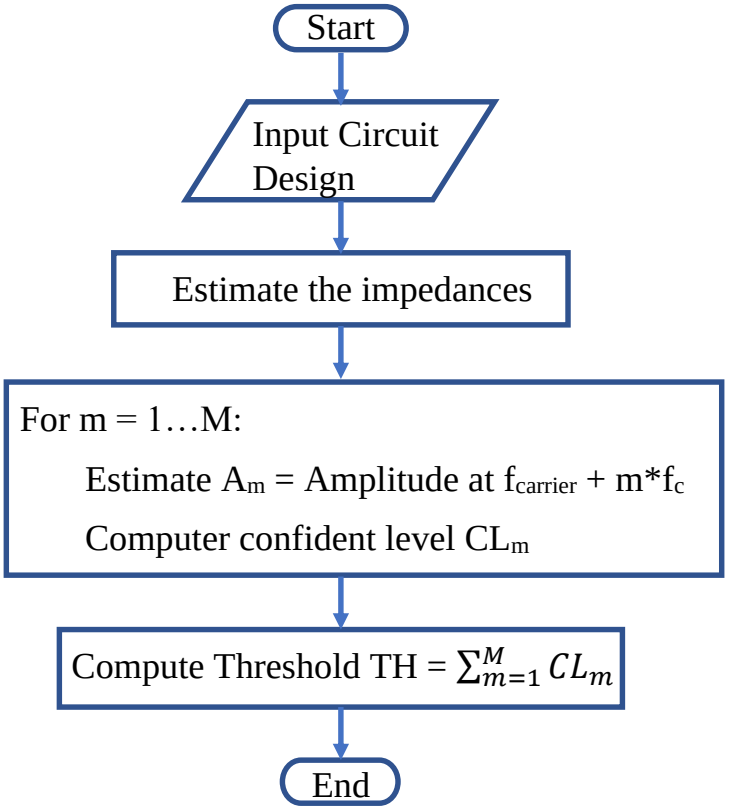


Figure 7.8: Estimation flow.

lower than this detection threshold. The summary of this detection phase is illustrated in Fig. 7.9.

7.5 Evaluation

7.5.1 Benchmark Implementation and Measurement Setup

We have implemented an 2^{20}-bit ripple adder circuit as illustrated in Fig. 7.10. We choose to test on this simple sequential circuit first so we can have better intuition on how effective our technique is and what is the limit before testing on real benchmarks. We inject a simple hardware Trojan, whose trigger takes N-bit from the output of the ripple circuit as inputs and get activated when a predefined sequence of values is detected. We change the number of bits N to change the trigger size of the Trojan. The payload is just a shift register that keeps switching once the Trojan gets activated.

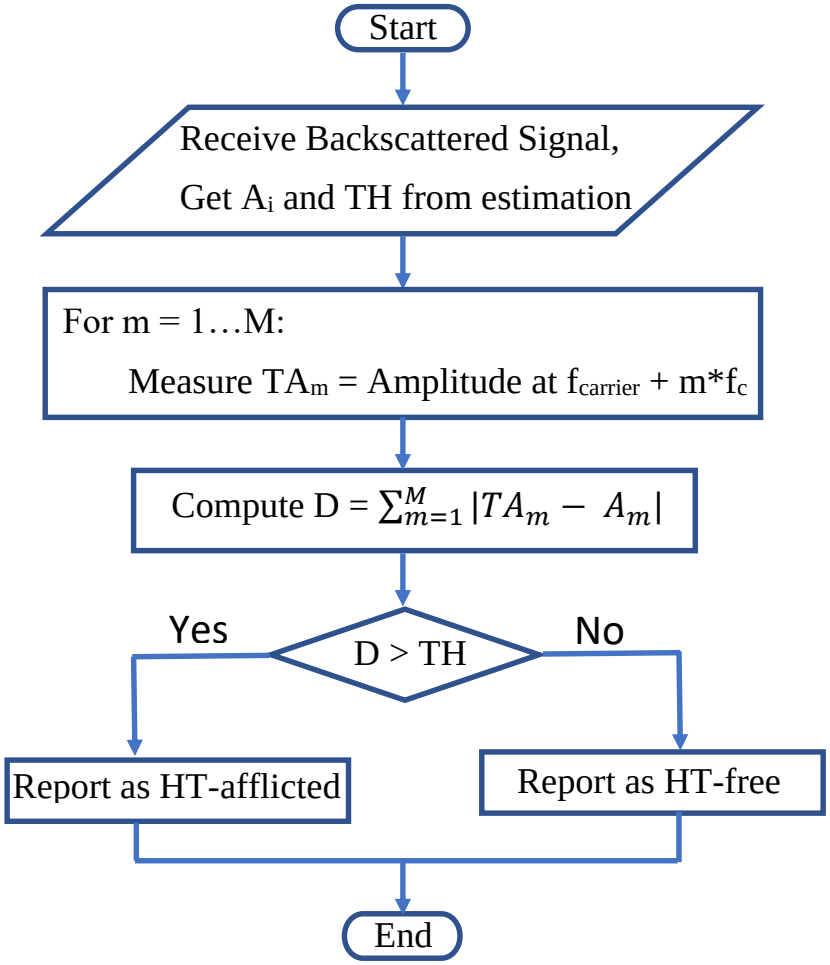


Figure 7.9: Detection flow.

The Trojan-affected and Trojan-free designs are carefully mapped to the FPGA by using ECO (Engineering Change Order) tools so that they have the same layout except for the Trojan part, thus making a fair comparison. The experimental setup to evaluate the performance of the proposed algorithm is shown in Fig. 5.7. The setup includes a transmitter Aaronia E1 electric-field near-field probe [103] connected to an Agilent MXG N5183A signal generator [108], and a receiver Aaronia H2 magnetic field near-field probe [103] connected to an Agilent MXA N9020A spectrum analyzer [109]. The devices-under-test (DuT) are Altera DE0 Cyclone V FPGA boards [110]. An angle ruler is used as a positioner so that different DE0-CV boards can be tested using approximately the same position of probes. A laptop is used to control the devices and automate the measurements. A 3 GHz

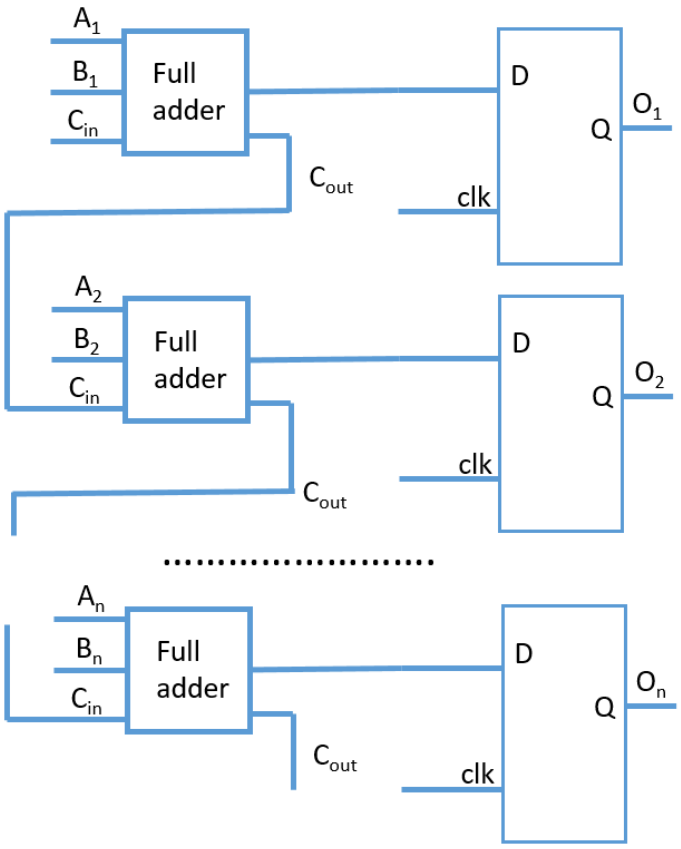


Figure 7.10: A ripple adder circuit.

Table 7.5: Summary of Trojan designs for the ripple adder circuit

Trojan design	Size (% of the original circuit)
TJ1	3.17%
TJ2	6.25%
TJ3	9.38%

continuous sinusoid signal is generated by the signal generator, and backscattered signals are recorded by the spectrum analyzer. Table 7.5 summarizes the Trojan benchmarks.

7.5.2 Results

We evaluate the effectiveness of our HT detection prototype by applying HT detection to four test subject circuits implemented on FPGA, including one original ripple adder design, and three Trojan-infected designs, in which each is infected by a Trojan in Table 7.5. The impedance model allows us to estimate the power of the backscattered signal up to the

third harmonics of the clock. We use these estimated values as the base to compare the experiment values with. Each test subject design is measured 20 times, and each measurement is used for HT detection in isolation, i.e., for each test subject, the detection makes 20 classification decisions (HT-free or HT-afflicted).

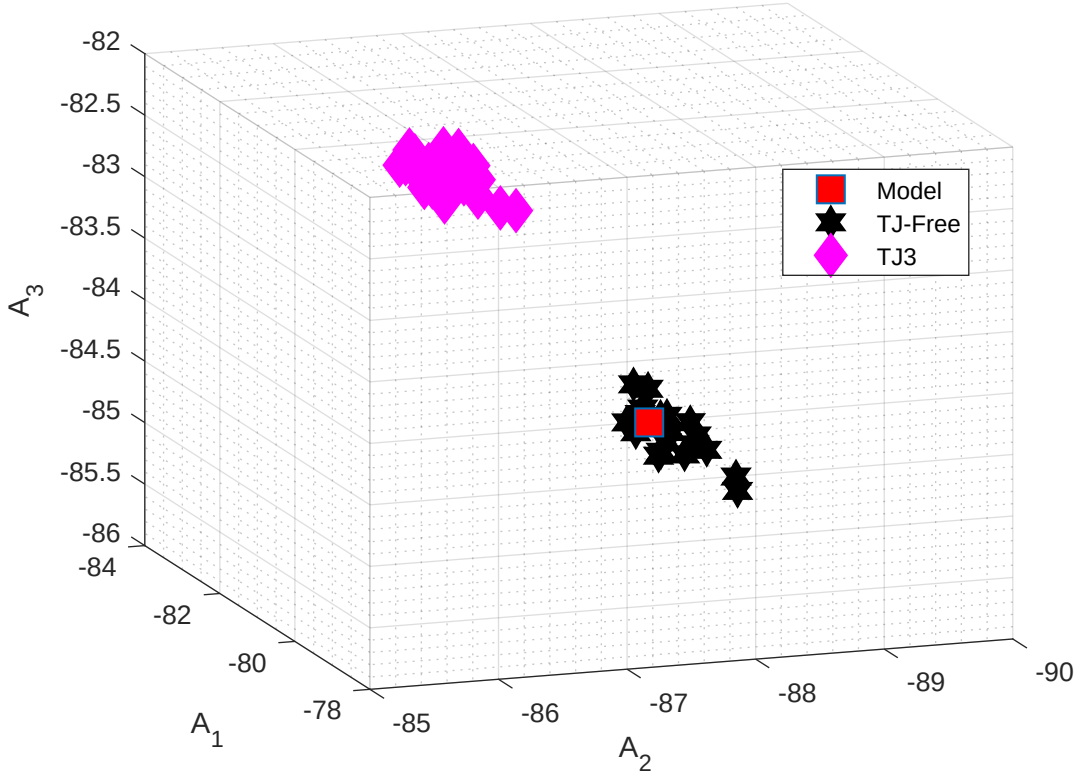


Figure 7.11: Estimation and measurements of backscattered signal power for the ripple adder and different Trojan-infected designs.

Fig. 7.11 shows estimation for the power of the backscattered signal of the clock harmonics of a ripple adder circuit, and measurements for HT-free ripple adder and the TJ3-infected ripple adder design. The figure shows that the estimation is pretty correct as the TJ-free measurements cluster around the estimation. The TJ3-infected design measurements are well separated from the TJ-free design measurements, which indicates that our algorithm would be able to correctly detect all the TJ3-infected design measurements as Trojan-infected ones. However, as the Trojan gets smaller in TJ1 and TJ2 designs, the measurements are not separated from the TJ-free ones. Fig. 7.12 shows the ROC curves when using our detection technique for these designs. The results show that we can detect

Trojan 3 with 100% and 0% false positives, however, as the Trojan gets smaller, we can only detect with 80% and 30% accuracy for Trojan 2, and 1, respectively if we want to keep the 0% false positives. Compared to the previous sections, this golden-chip-free technique requires the Trojan to be significantly bigger to be detected. This is because we only use three clock harmonics in the model.

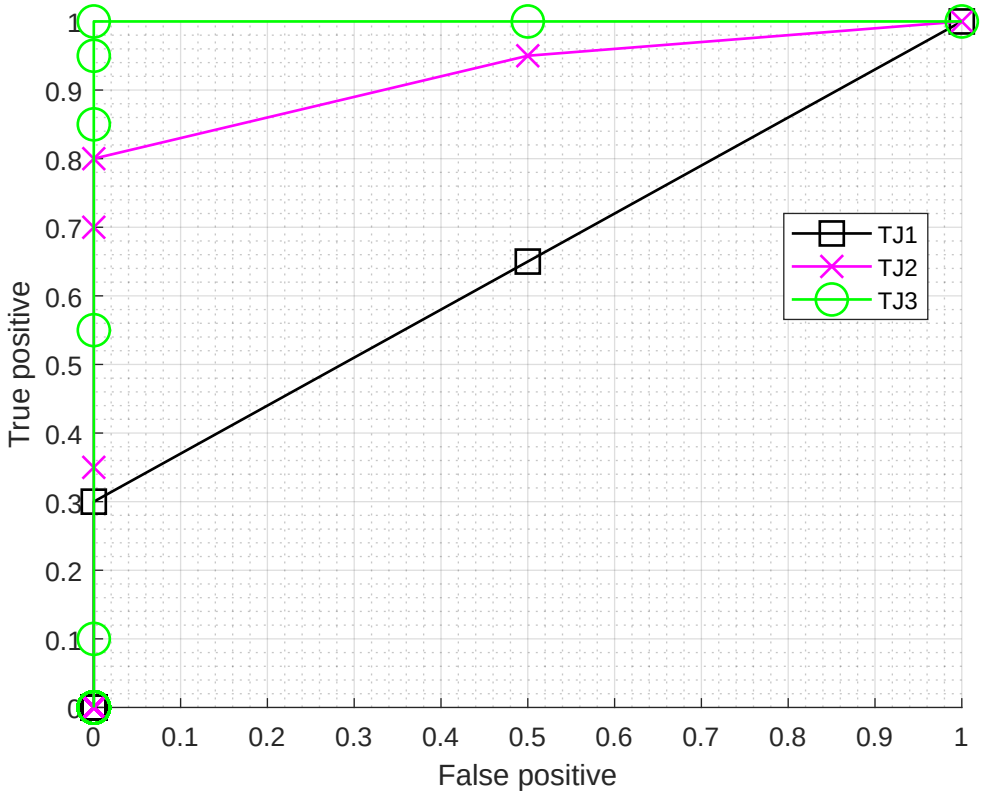


Figure 7.12: ROC curves for different Trojan-infected ripple circuit design.

7.6 Further Evaluation on Real Benchmark Circuits

To further evaluate the effectiveness of our HT detection prototype, we use the RS232 circuit with three HTs, from TrustHub. We choose the RS232 benchmark because its Trojan designs can be easily increased in size. We implement two different extended versions of RS232-T500 and a extended version of RS232-T300, whose trigger sizes are increased bigger that the original one's. Table summarizes the test designs we use.

Table 7.6: Summary of Trojan designs for the RS232 circuit

Trojan design	Size (% of the original circuit)	
	Trigger	Payload
RS232-T500-ver1	10.02%	1.48%
RS232-T300-ver1	9.56%	1.58%
RS232-T500-ver2	5.85%	1.48%

- RS232-T500: The payload in this HT is a circuit that, upon activation, causes the transmission to fail. The trigger is a sequential circuit that increments its counter every clock cycle and activates the payload when this counter reaches a certain value.

- RS232-T300: The payload in this HT is a circuit that, upon activation, gains control over two primary output signals. The trigger is a sequential comparator whose trigger input probability is $8e^{-20}$.

We repeat the same experiments as in Section 7.5 for these test designs. The results in Fig. 7.13 show that the expected values estimated by the model are accurate as the TJ-free measurements are close to the model. This is similar to what we have observed in the previous section. Both RS232-T500-ver1 and RS232-T300-ver1 measurements are separated from the model and the TJ-free design measurements, which means that the algorithm can detect them easily. However, as the Trojan gets smaller, the measurements are not separated from the TJ-free ones, which means it is harder for the algorithm to detect them. Fig. 7.14 shows the ROC curves when using our detection technique for these designs. The results show that we can detect both RS232-T500-ver1 and RS232-T300-ver1 with 100% and 0% false positives, however, as the Trojan gets smaller, the accuracy of the technique significantly drops. This also matches with what we have observed in Section 7.5.

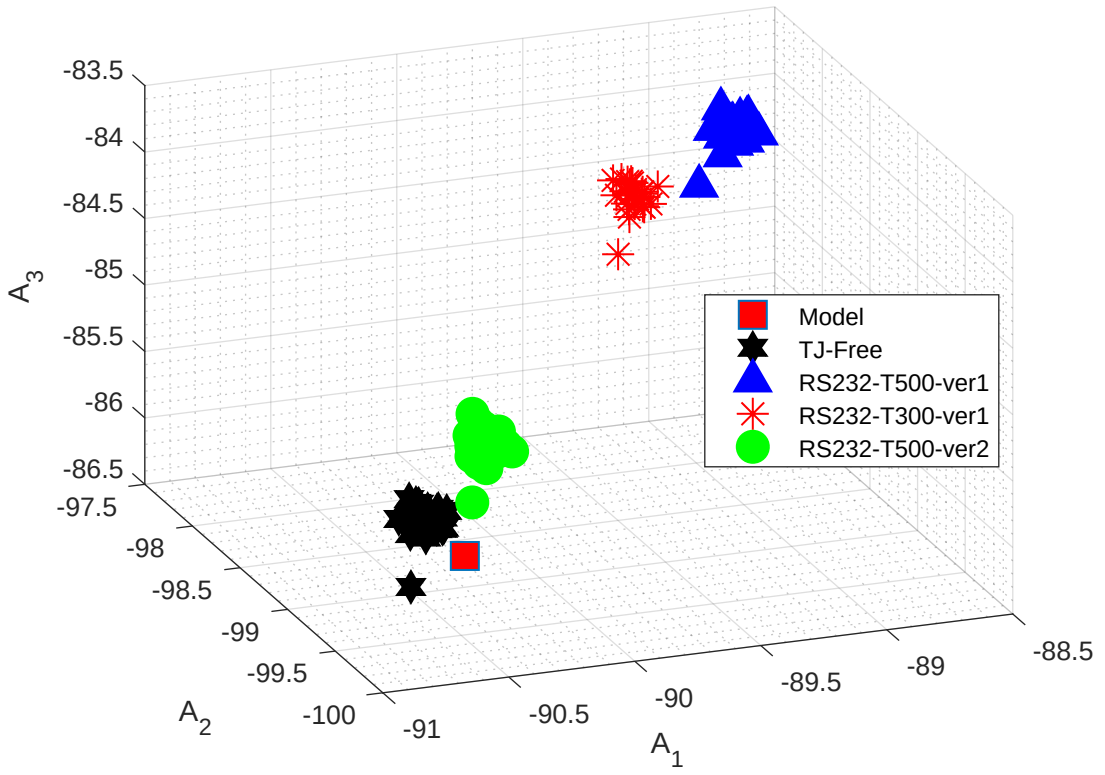


Figure 7.13: Estimation and measurements of backscattered signal power for the rs232 and different Trojan-infected designs.

7.7 Conclusions

Existing side-channel analysis based HT detection techniques rely on the assumption of having a golden (HT-free) chip for training. This assumption of having a golden sample is too strong, and often unrealistic, which prevents them from being used for practical deployments of HT detection. This chapter tackles the problem by proposing a novel golden-chip-free hardware Trojan detection technique using backscattering side-channel with circuit's impedance modeling. We test our technique on multiple Trojan benchmarks and the results show that our technique can detect Trojan with 100% accuracy and 0% false positives, if the Trojan trigger is big enough.

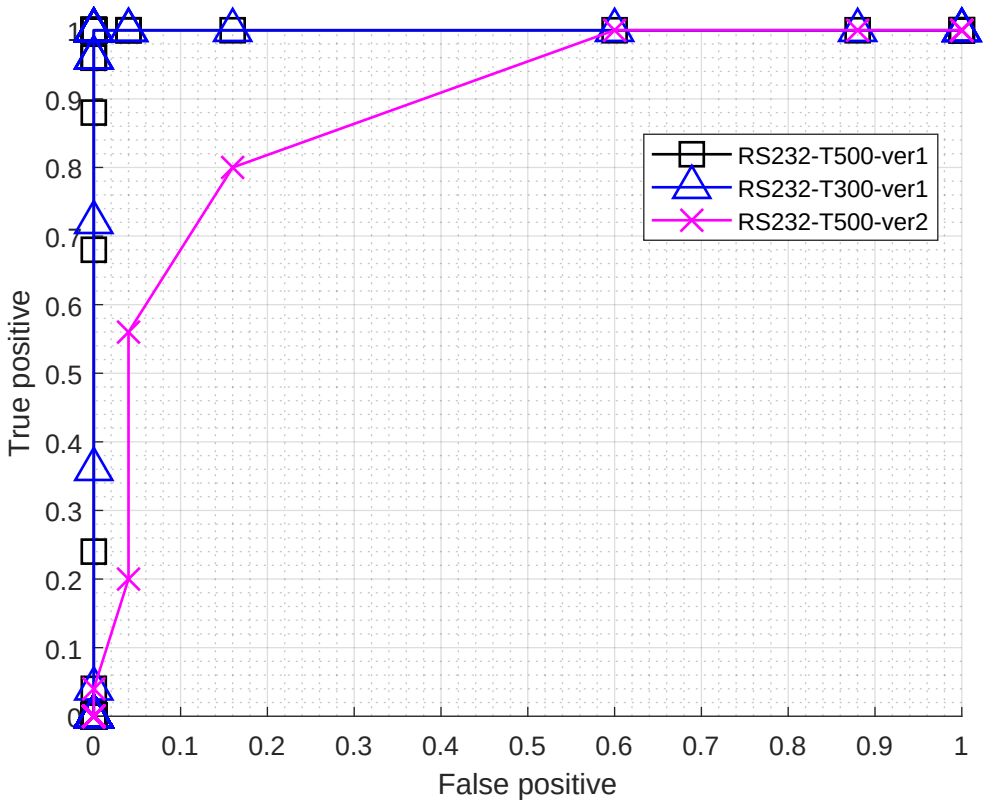


Figure 7.14: ROC curves for different Trojan-infected RS232 designs.

CHAPTER 8
CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

This research introduces a new physical side-channel, which we call the backscattering side-channel, and propose novel hardware Trojan (HT) and counterfeit integrated circuit (IC) detection techniques that exploit the backscattering side channel. Backscattering has been used RFID communication system to enable RFID tags to transmit information to RFID reader for decades, but it has never been used as a side-channel before this work. The backscattering side-channel is a consequence of impedance changes in switching circuits, which is caused by the transistors' two-state impedances modulating and reflecting an injected carrier signal. As a result, this new side-channel is supposed to work well for detecting malicious changes at the circuitry level of IC. This is proved by the results that our proposed hardware Trojan (HT) and counterfeit integrated circuit (IC) detection techniques using backscattering side-channel are capable of detecting different types of inactive HTs and counterfeit ICs on multiple circuit benchmarks while tolerating manufacturing variation, and the backscattering side-channel outperforms other side-channels, such as EM and power side-channels, when using the same detection prototypes. The research contributions of this work are:

- Discover the backscattering side-channel, a new side-channel that is created by transmitting a signal toward the IC, where the internal impedance changes caused by on-chip switching activity change the circuit's RCS, thus modulate the signal that is backscattered (reflected) from the from the circuit with the information about impedance changes in the system. Unlike other analog side-channels such as electromagnetic emanation (EM) and power, which are a consequence of current-flow

changes inside the chip, backscattering side-channel is an impedance-based side channel that is the consequence of impedance switching activities inside the chip. If hardware Trojan is added to a circuit, it changes the impedance of the circuit even if the Trojan is not activated. The changes will be reflected in the backscattered signal, which is beneficial to the detection of hardware Trojan. The same logic can be applied for the detection of counterfeit ICs. The backscattering side-channel has several advantages compared to other side-channels such as EM and power. These advantages can be listed as follows:

  - *High bandwidth:* This provides the capability of detecting small and fast switching Trojan activities.
  - *Signal strength not limited by leakage from devices:* One characteristic that sets the backscattering side-channel aside from others is that its signal strength can be improved by increasing the carrier's input power. As a result, the backscattering side-channel can still work when there is very little leakage from devices.
  - *Adaptable frequency:* By changing the carrier frequency, we can change the working frequency of the backscattering side-channel. This helps to increase the signal-to-noise ratio by shifting the frequency to avoid interrupts that might distract the changes caused by HT activities.

- Propose novel techniques for the detection of hardware Trojan and counterfeit IC using the new backscattering side-channels. These techniques rely analyzing impedance changes within sub-clock samples, where the changes caused by HTs happen and can be observed on the clock signal, to detect malicious changes at the circuitry level. They depends on having a golden (Trojan-free) chip to generate reference signals for detecting malicious modifications (if existed) in the chip circuitry. To our knowledge, this is the first off-chip side-channel technique capable of detecting *inactive* HTs and counterfeit ICs while tolerating variations that exist across hardware instances. We

experimentally confirm, using measurements on one physical instance for training and nine other physical instances for testing, that the new techniques allow detection of a dormant HT and counterfeit ICs with 100% accuracy, while producing no false positives in HT-free measurements. Furthermore, additional experiments are conducted to compare the backscattering-based detection to one that uses the traditional EM-emanation-based side channel. These results show that backscattering-based detection outperforms the EM side channel, confirm that dormant HTs are much more difficult for detection than HTs that have been activated, and show how detection is affected by changing the HT's size and physical location on the IC.

- Model and quantitatively compare backscattering, electromagnetic (EM), and power side-channels and discuss the performance of these three side-channels for detecting software malware and hardware Trojans. Side-channel analysis is a powerful tool both from attacker's and from defender's perspectives. Understanding similarities and differences among a large number of side-channels is a necessary step in better utilizing them. This work addresses this problem by modeling and quantitatively comparing the backscattering, EM, and power side-channels and discussing the performance of these three side-channels in detecting software malware and HT. We proved that for larger changes in the signals, such as those caused by malware intrusions, all three side-channels perform similarly. However, when smaller changes need to be observed, such as those caused by HTs, backscattering side-channel outperforms EM and power side-channels.

- Propose a novel clustering algorithm that is capable of classifying a large population of ICs into clusters without having a "golden" (known-to-be-HT-free) chip, and with no a priori knowledge about circuitry of the chip. The technique bridges the gap between two existing hardware Trojan detection paradigms: accurate, but destructive, expensive and time-consuming reverse engineering and cheap, fast and nondestruc-

tive but goden-chip dependent side-channel analysis. This work proposes a novel golden-chip-free clustering method using backscattering side-channel to divide ICs into groups of Trojan-free and Trojan-infected boards. The technique requires no golden chip or a priori knowledge of the chip circuitry, and divides a large population of ICs into clusters based on how HTs (if existed) affect their backscattered signals. This significantly reduces the size of test vectors for reverse engineering based detection techniques, thus enables deployment of reverse engineering approaches to a large population of ICs in a real testing scenario. We tested the proposed algorithm on a set of 100 boards to thoroughly evaluate manufacturing variations among different hardware instances. The results showed that our technique can tolerate manufacturing variations among hardware instances to cluster all boards correctly for not only 9 different dormant Trojan designs on 3 different benchmark circuits from Trusthub, but also dormant Trojan designs whose trigger size is as small as 0.19% of the original circuit.

- Propose models to estimate impedances of a circuit and power of the backscattered signal of clock harmonics, and introduce a golden-chip-free hardware Trojan detection technique using these models.

8.2 Future Work

8.2.1 Exploiting High Spatial Resolution of Backscattering Side-Channel for Hardware Trojan Detection

One of the advantages of backscattering side-channel is that it offers high spacial resolution, which means it can focus on specific part of the chip. However, the probes we currently use are almost as big as the whole chip, which prevents us from exploiting the high spacial resolution of backscattering side-channel. With a smaller probe, techniques can be developed to scan through the chip and to investigate if it helps to improve the detection accuracy and

give information about the location of HTs in the chip.

8.2.2 Improving Golden-Chip Free Hardware Trojan Detection Techniques

As discussed throughout this thesis, one of the main drawbacks of existing side-channel based HT detection techniques is the dependence on having a golden (HT-free) sample. This thesis proposed novel golden free HT detection techniques by modeling the circuit to estimate its impedance, and estimating the power of the backscattering side-channel signal. The results show that our techniques are capable of detecting Trojans with 100% accuracy and 0% false positive if the Trojan is big enough. The model needs to be refined so it can model higher clock harmonics' power, which would allow for smaller hardware Trojan detection.

REFERENCES

[1] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M Tehranipoor, "Hardware trojans: Lessons learned after one decade of research," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 22, no. 1, p. 6, 2016.

[2] W. K. Clark and P. L. Levin, "Securing the information highway," *Foreign Aff.*, vol. 88, p. 2, 2009.

[3] J. Villasenor, *Compromised by design?: Securing the defense electronics supply chain*. Center for Technology Innovation at Brookings, 2013.

[4] ——, "The hacker in your hardware," *Scientific American*, vol. 303, no. 2, pp. 82–87, 2010.

[5] L.-W. Kim, J. D. Villasenor, *et al.*, "A trojan-resistant system-on-chip bus architecture," in *Military Communications Conference, 2009. MILCOM 2009. IEEE*, IEEE, 2009, pp. 1–6.

[6] Q. Yu and J. Frey, "Exploiting error control approaches for hardware trojans on network-on-chip links," in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2013 IEEE International Symposium on*, IEEE, 2013, pp. 266–271.

[7] D McIntyre, F Wolff, C Papachristou, S. Bhunia, and D Weyer, "Dynamic evaluation of hardware trust," in *Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on*, IEEE, 2009, pp. 108–111.

[8] L.-W. Kim and J. D. Villasenor, "Dynamic function replacement for system-on-chip security in the presence of hardware-based attacks," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 661–675, 2014.

[9] R. Torrance and D. James, "The state-of-the-art in ic reverse engineering," in *Cryptographic Hardware and Embedded Systems-CHES 2009*, Springer, 2009, pp. 363–381.

[10] A. Waksman, M. Suozzo, and S. Sethumadhavan, "Fanci: Identification of stealthy malicious logic using boolean functional analysis," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, ACM, 2013, pp. 697–708.

[11] H. Salmani, "Cotd: Reference-free hardware trojan detection and recovery based on controllability and observability in gate-level netlist," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 338–350, 2017.

[12] J. Zhang, F. Yuan, L. Wei, Y. Liu, and Q. Xu, "Veritrust: Verification for hardware trust," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 7, pp. 1148–1161, 2015.

[13] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE design & test of computers*, vol. 27, no. 1, 2010.

[14] R. S. Chakraborty, S. Narasimhan, and S. Bhunia, "Hardware trojan: Threats and emerging solutions," in *High Level Design Validation and Test Workshop, 2009. HLDVT 2009. IEEE International*, IEEE, 2009, pp. 166–171.

[15] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using ic fingerprinting," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*, IEEE, 2007, pp. 296–310.

[16] M. Banga and M. S. Hsiao, "A region based approach for the identification of hardware trojans," in *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, IEEE, 2008, pp. 40–47.

[17] ——, "Vitamin: Voltage inversion technique to ascertain malicious insertions in ics," 2009.

[18] C. He, B. Hou, L. Wang, Y. En, and S. Xie, "A failure physics model for hardware trojan detection based on frequency spectrum analysis," in *Reliability Physics Symposium (IRPS), 2015 IEEE International*, IEEE, 2015, PR–1.

[19] S. Narasimhan, D. Du, R. S. Chakraborty, S. Paul, F. Wolff, C. Papachristou, K. Roy, and S. Bhunia, "Multiple-parameter side-channel analysis: A non-invasive hardware trojan detection approach," in *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on*, IEEE, 2010, pp. 13–18.

[20] C. Bao, D. Forte, and A. Srivastava, "Temperature tracking: Toward robust run-time detection of hardware trojans," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1577–1585, 2015.

[21] D. Forte, C. Bao, and A. Srivastava, "Temperature tracking: An innovative run-time approach for hardware trojan detection," in *Proceedings of the International Conference on Computer-Aided Design*, IEEE Press, 2013, pp. 532–539.

[22] J. He, Y. Zhao, X. Guo, and Y. Jin, "Hardware trojan detection through chip-free electromagnetic side-channel statistical analysis," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2939–2948, 2017.

[23] J. Balasch, B. Gierlichs, and I. Verbauwhede, "Electromagnetic circuit fingerprints for hardware trojan detection," in *Electromagnetic Compatibility (EMC), 2015 IEEE International Symposium on*, IEEE, 2015, pp. 246–251.

[24] X. T. Ngo, Z. Najm, S. Bhasin, S. Guilley, and J.-L. Danger, "Method taking into account process dispersion to detect hardware trojan horse by side-channel analysis," *Journal of Cryptographic Engineering*, vol. 6, no. 3, pp. 239–247, 2016.

[25] K. Hu, A. N. Nowroz, S. Reda, and F. Koushanfar, "High-sensitivity hardware trojan detection using multimodal characterization," in *Proceedings of the Conference on Design, Automation and Test in Europe*, EDA Consortium, 2013, pp. 1271–1276.

[26] A. N. Nowroz, K. Hu, F. Koushanfar, and S. Reda, "Novel techniques for high-sensitivity hardware trojan detection using thermal and power maps," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1792–1805, 2014.

[27] B. Cha and S. K. Gupta, "Efficient trojan detection via calibration of process variations," in *Test Symposium (ATS), 2012 IEEE 21st Asian*, IEEE, 2012, pp. 355–361.

[28] ——, "Trojan detection via delay measurements: A new approach to select paths and vectors to maximize effectiveness and minimize cost," in *Proceedings of the conference on design, automation and test in Europe*, EDA Consortium, 2013, pp. 1265–1270.

[29] M. Lecomte, J. Fournier, and P. Maurine, "An on-chip technique to detect hardware trojans and assist counterfeit identification," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 12, pp. 3317–3330, 2017.

[30] P. V. Nikitin and K. S. Rao, "Theory and measurement of backscattering from rfid tags," *IEEE Antennas and Propagation Magazine*, vol. 48, no. 6, pp. 212–218, 2006.

[31] B. Shakya, T. He, H. Salmani, D. Forte, S. Bhunia, and M. Tehranipoor, "Benchmarking of hardware trojans and maliciously affected circuits," *Journal of Hardware and Systems Security*, vol. 1, no. 1, pp. 85–102, 2017.

[32] P Kocher, J Jaffe, and B Jun, "Differential power analysis: leaking secrets," in *Proceedings of CRYPTO'99, Springer, Lecture notes in computer science*, 1999, pp. 388–397.

[33] A. G. Bayrak, F Regazzoni, P Brisk, F.-X. Standaert, and P Ienne, "A first step towards automatic application of power analysis countermeasures," in *Proceedings of the 48th Design Automation Conference (DAC)*, 2011.

[34] U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, M. Majzoobi, F. Koushanfar, and W. Burleson, "Efficient power and timing side channels for physical unclonable functions," in *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2014, pp. 476–492.

[35] M Backes, M Durmuth, S Gerling, M Pinkal, and C Sporleder, "Acoustic side-channel attacks on printers," in *Proceedings of the USENIX Security Symposium*, 2010.

[36] S Chari, J. R. Rao, and P Rohatgi, "Template attacks," in *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2002*, 2002, pp. 13–28.

[37] D Agrawal, B Archambeult, J. R. Rao, and P Rohatgi, "The EM side-channel(s)," in *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2002*, 2002, pp. 29–45.

[38] D. Genkin, I. Pipman, and E. Tromer, "Get your hands off my laptop: Physical side-channel key-extraction attacks on pcs," *Journal of Cryptographic Engineering*, vol. 5, no. 2, pp. 95–112, 2015.

[39] M. Alam, H. A. Khan, M. Dey, N. Sinha, R. L. Callan, A. G. Zajic, and M. Prvulovic, "One&done: A single-decryption EM-based attack on OpenSSL's constant-time blinded RSA," in *USENIX Security*, 2018, pp. 585–602.

[40] H. Sekiguchi and S. Seto, "Study on maximum receivable distance for radiated emission of information technology equipment causing information leakage," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, no. 3, pp. 547–554, 2013.

[41] Y.-i. Hayashi, N. Homma, T. Mizuki, H. Shimada, T. Aoki, H. Sone, L. Sauvage, and J.-L. Danger, "Efficient evaluation of em radiation associated with information leakage from cryptographic devices," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, no. 3, pp. 555–563, 2013.

[42] K Gandolfi, C Mourtel, and F Olivier, "Electromagnetic analysis: concrete results," in *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2001*, 2001, pp. 251–261.

[43] M. Vuagnoux and S. Pasini, "An improved technique to discover compromising electromagnetic emanations," in *2010 IEEE International Symposium on Electromagnetic Compatibility*, 2010, pp. 121–126.

[44] Y. I. Hayashi, N. Homma, T. Mizuki, T. Aoki, H. Sone, L. Sauvage, and J. L. Danger, "Analysis of electromagnetic information leakage from cryptographic devices with different physical structures," *IEEE Transactions on Electromagnetic Compatibility*, vol. 55, no. 3, pp. 571–580, 2013.

[45] R. Callan, A. Zajic, and M. Prvulovic, "A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events," in *Microarchitecture (MICRO), 2014 47th Annual IEEE/ACM International Symposium on*, IEEE, 2014, pp. 242–254.

[46] B. B. Yilmaz, R. L. Callan, M. Prvulovic, and A. Zajić, "Capacity of the em covert/side-channel created by the execution of instructions in a processor," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 605–620, 2017.

[47] B. B. Yilmaz, M. Prvulovic, and A. Zajić, "Electromagnetic side channel information leakage created by execution of series of instructions in a computer processor," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 776–789, 2019.

[48] B. B. Yilmaz, N. Sehatbakhsh, A. Zajić, and M. Prvulovic, "Communication model and capacity limits of covert channels created by software activities," *IEEE Transactions on Information Forensics and Security*, 2019.

[49] L. Liu, G. Yan, X. Zhang, and S. Chen, "Virusmeter: Preventing your cellphone from spies," in *International Workshop on Recent Advances in Intrusion Detection*, Springer, 2009, pp. 244–264.

[50] C. R. A. González and J. H. Reed, "Power fingerprinting in sdr integrity assessment for security and regulatory compliance," *Analog Integrated Circuits and Signal Processing*, vol. 69, no. 2-3, p. 307, 2011.

[51] S. S. Clark, B. Ransford, A. Rahmati, S. Guineau, J. Sorber, W. Xu, and K. Fu, "Wattsupdoc: Power side channels to nonintrusively discover untargeted malware on embedded medical devices.," in *HealthTech*, 2013.

[52] R. Callan, F. Behrang, A. Zajic, M. Prvulovic, and A. Orso, "Zero-overhead profiling via em emanations," in *Proceedings of the 25th International Symposium on Software Testing and Analysis*, ACM, 2016, pp. 401–412.

[53] A. Nazari, N. Sehatbakhsh, M. Alam, A. Zajic, and M. Prvulovic, "Eddie: Em-based detection of deviations in program execution," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, ser. ISCA '17, Toronto, ON, Canada, 2017, pp. 333–346, ISBN: 978-1-4503-4892-8.

[54] H. A. Khan, N. Sehatbakhsh, L. N. Nguyen, R. Callan, A. Yeredor, M. Prvulovic, and A. Zajić, "Idea: Intrusion detection through electromagnetic-signal analysis for critical embedded and cyber-physical systems," *IEEE Transactions on Dependable and Secure Computing (to be published)*, 2019.

[55] H. A. Khan, N. Sehatbakhsh, L. N. Nguyen, M. Prvulovic, and A. Zajić, "Malware detection in embedded systems using neural network model for electromagnetic side-channel signals," *Journal of Hardware and Systems Security (to be published)*, 2019.

[56] R. Callan, A. Zajić, and M. Prvulovic, "Fase: Finding amplitude-modulated side-channel emanations," in *ACM SIGARCH Computer Architecture News*, ACM, vol. 43, 2015, pp. 592–603.

[57] B. B. Yilmaz, E. M. Ugurlu, A. Zajic, and M. Prvulovic, "Instruction level program tracking using electromagnetic emanations," in *Proceedings of the SPIE*, International Society for Optics and Photonics, vol. 11011, 2019.

[58] N. Sehatbakhsh, A. Nazari, A. Zajic, and M. Prvulovic, "Spectral profiling: Observer-effect-free profiling by monitoring em emanations," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2016, pp. 1–11.

[59] H. A. Khan, M. Alam, A. Zajic, and M. Prvulovic, "Detailed tracking of program control flow using analog side-channel signals: A promise for iot malware detection and a threat for many cryptographic implementations," in *Cyber Sensing 2018*, International Society for Optics and Photonics, vol. 10630, 2018, p. 1 063 005.

[60] N. Sehatbakhsh, M. Alam, A. Nazari, A. Zajic, and M. Prvulovic, "Syndrome: Spectral analysis for anomaly detection on medical iot and embedded devices," in *2018 IEEE international symposium on hardware oriented security and trust (HOST)*, IEEE, 2018, pp. 1–8.

[61] M. Dey, A. Nazari, A. Zajic, and M. Prvulovic, "Emprof: Memory profiling via em-emanation in iot and hand-held devices," in *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, IEEE, 2018, pp. 881–893.

[62] N. Sehatbakhsh, A. Nazari, H. Khan, A. Zajic, and M. Prvulovic, "Emma: Hardware/software attestation framework for embedded systems using electromagnetic signals," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2019, pp. 983–995.

[63] N. Sehatbakhsh, A. Nazari, M. Alam, F. Werner, Y. Zhu, A. Zajic, and M. Prvulovic, "Remote: Robust external malware detection framework by using electromagnetic signals," *IEEE Transactions on Computers*, 2019.

[64] L. N. Nguyen, C. Cheng, M. Prvulovic, and A. Zajic, "Creating a backscattering side channel to enable detection of dormant hardware trojans," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 7, pp. 1561–1574, 2019.

[65] A. A. Nasr and M. Z. Abdulmageed, "An efficient reverse engineering hardware trojan detector using histogram of oriented gradients," *Journal of Electronic Testing*, vol. 33, no. 1, pp. 93–105, 2017.

[66] M. Fyrbiak, S. Wallat, P. Swierczynski, M. Hoffmann, S. Hoppach, M. Wilhelm, T. Weidlich, R. Tessier, and C. Paar, "Hal—the missing piece of the puzzle for hardware reverse engineering, trojan detection and insertion," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 3, pp. 498–510, 2018.

[67] C. Bao, D. Forte, and A. Srivastava, "On reverse engineering-based hardware trojan detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 49–57, 2016.

[68] S. Wallat, M. Fyrbiak, M. Schlögel, and C. Paar, "A look at the dark side of hardware reverse engineering - a case study," in *2017 IEEE 2nd International Verification and Security Workshop (IVSW)*, 2017, pp. 95–100.

[69] C. Bao, D. Forte, and A. Srivastava, "On application of one-class svm to reverse engineering-based hardware trojan detection," in *Fifteenth International Symposium on Quality Electronic Design*, 2014, pp. 47–54.

[70] X. Wei, Y. Diao, and Y. Wu, "To detect, locate, and mask hardware trojans in digital circuits by reverse engineering and functional eco," in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 623–630.

[71] R. Vaikuntapu, L. Bhargava, and V. Sahula, "Golden ic free methodology for hardware trojan detection using symmetric path delays," in *2016 20th International Symposium on VLSI Design and Test (VDAT)*, 2016, pp. 1–2.

[72] Y. Tang, S. Li, L. Fang, X. Hu, and J. Chen, "Golden-chip-free hardware trojan detection through quiescent thermal maps," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 1–12, 2019.

[73] L. N. Nguyen, C. Cheng, M. Prvulovic, and A. Zajić, "Creating a backscattering side channel to enable detection of dormant hardware trojans," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 7, pp. 1561–1574, 2019.

[74] B. Hou, C. He, L. Wang, Y. En, and S. Xie, "Hardware trojan detection via current measurement: A method immune to process variation effects," in *2014 10th International Conference on Reliability, Maintainability and Safety (ICRMS)*, 2014, pp. 1039–1042.

[75] B. Çakir and S. Malik, "Hardware trojan detection for gate-level ics using signal correlation based clustering," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, EDA Consortium, 2015, pp. 471–476.

[76] P.-S. Ba, S. Dupuis, M.-L. Flottes, G. Di Natale, and B. Rouzeyre, "Using outliers to detect stealthy hardware trojan triggering?" In *Verification and Security Workshop (IVSW), IEEE International*, IEEE, 2016, pp. 1–6.

[77] M. Xue, R. Bian, W. Liu, and J. Wang, "Defeating untrustworthy testing parties: A novel hybrid clustering ensemble based golden models-free hardware trojan detection method," *IEEE Access*, 2018.

[78] A. Basak, Y. Zheng, and S. Bhunia, "Active defense against counterfeiting attacks through robust antifuse-based on-chip locks," in *Proc. IEEE 32nd VLSI Test Symposium (VTS)*, 2014, pp. 1–6.

[79] R. Moudgil, D. Ganta, L. Nazhandali, M. Hsiao, C. Wang, and S. Hall, "A novel statistical and circuit-based technique for counterfeit detection in existing ics," in *Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI*, ACM, 2013, pp. 1–6.

[80] K. Mahmood, P. L. Carmona, S. Shahbazmohamadi, F. Pla, and B. Javidi, "Real-time automated counterfeit integrated circuit detection using x-ray microscopy," *Applied Optics*, vol. 54, no. 13, pp. D25–D32, 2015.

[81] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, 2014.

[82] K. Yang, D. Forte, and M. Tehranipoor, "An rfid-based technology for electronic component and system counterfeit detection and traceability," in *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, IEEE, 2015, pp. 1–6.

[83] Y. Zheng, A. Mannai, and M. Sawan, "A biomems chip with integrated micro electromagnet array towards bio-particles manipulation," *Microelectronic Engineering*, vol. 128, pp. 1–6, 2014.

[84] M. M. Tehranipoor, U. Guin, and D. Forte, "Counterfeit integrated circuits," in *Counterfeit Integrated Circuits*, Springer, 2015, pp. 15–36.

[85] K. He, X. Huang, and S. X.-D. Tan, "Em-based on-chip aging sensor for detection and prevention of counterfeit and recycled ics," in *Proc. IEEE Int'l. Conf. on Computer-Aided Design*, 2015, pp. 146–151.

[86] P. Song, F. Stellari, and A. Weger, "Counterfeit ic detection using light emission," in *IEEE International Test Conference*, 2014, pp. 1–8.

[87] P. Ghosh and R. S. Chakraborty, "Counterfeit ic detection by image texture analysis," in *Euromicro conference on digital system design (DSD)*, 2017, pp. 283–286.

[88] A. H. Baba and S. Mitra, "Testing for transistor aging," in *27th IEEE VLSI Test Symp.*, 2009, pp. 215–220.

[89] X. Chen, Y. Wang, Y. Cao, Y. Ma, and H. Yang, "Variation-aware supply voltage assignment for minimizing circuit degradation and leakage," in *Proceedings of the 2009 ACM/IEEE international symposium on Low power electronics and design*, ACM, 2009, pp. 39–44.

[90] F. T.W.M.P.A. Z. Luong N. Nguyen Chia-Lin Cheng, "A comparison of backscattering, em, and power side- channels and their performance in detecting software and hardware intrusions," *Journal of Hardware and Systems Security*, 2020.

[91] M. P. Luong N. Nguyen Baki Berkay Yilmaz and A. Zajić, "A novel golden-chip-free clustering technique using backscattering side channel for hardware trojan detection," in *Hardware-Oriented Security and Trust, 2020. HOST 2020. IEEE International Workshop on*, IEEE, 2020.

[92] C.-L.C.F.T.W.M.P.A. Z. Luong N. Nguyen Chia-Lin Cheng, "A novel clustering technique using backscattering side channel for counterfeit ic detection," in *Security and Defense 2020*, International Society for Optics and Photonics, 2020.

[93] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware trojan attacks: Threat analysis and countermeasures," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1229–1247, 2014.

[94] J. Zhang, F. Yuan, and Q. Xu, "Detrust: Defeating hardware trust verification with stealthy implicitly-triggered hardware trojans," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2014, pp. 153–166.

[95] Z. Chen, X. Guo, R. Nagesh, A. Reddy, M. Gora, and A. Maiti, "Hardware trojan designs on basys fpga board," *Embedded system challenge contest in cyber security awareness week-CSAW*, 2008.

[96] R. S. Chakraborty, I. Saha, A. Palchaudhuri, and G. K. Naik, "Hardware trojan insertion by direct modification of fpga configuration bitstream," *IEEE Design & Test*, vol. 30, no. 2, pp. 45–54, 2013.

[97] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting malicious inclusions in secure hardware: Challenges and solutions," in *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, IEEE, 2008, pp. 15–19.

[98] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *Computer*, vol. 43, no. 10, pp. 39–46, 2010.

[99] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em side channel (s)," in *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2002, pp. 29–45.

[100] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*, Springer, 1999, pp. 388–397.

[101] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits*. Prentice hall Englewood Cliffs, 2002, vol. 2.

[102] *Trusthub*, http://www.trust-hub.org/benchmarks/trojan.

[103] AARONIA PBS, http://www.aaronia.com/products/antennas/Near-Field-Probe-Set-PBS2..

[104] Keysight Power Probe, http://www.keysight.com/en/pd-2471132-pn-N7020A/..

[105] *Hornatenna*, https://www.com-power.com/ah118_horn_antenna.html.

[106] M. R. Guthaus, J. S. Pingenberg, D. Emst, T. M. Austin, T. Mudge, and R. B. Brown, "Mibench: A free, commercially representative embedded benchmark suite," in *Proceedings of the Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop*, ser. WWC '01, 2001.

[107] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. 2009.

[108] Keysight Signal Generator, https://www.keysight.com/en/pdx-x201724-pn-N5183A/mxg-microwave-analog-signal-generator-100-khz-to-40-ghz?pm=spc&nid=-32490.1150253&cc=US&lc=eng..

[109] Keysight Signal Analyzer, https://www.keysight.com/en/pdx-x202266-pn-N9020A/mxa-signal-analyzer-10-hz-to-265-ghz?pm=spc&nid=-32508.1150426&cc=US&lc=eng..

[110] DE1 FPGA on NIOS Processor, https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=921&PartNo=2..

[111] C. Bao, D. Forte, and A. Srivastava, "On reverse engineering-based hardware trojan detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 49–57, 2016.

[112] A. Kulkarni, Y. Pino, and T. Mohsenin, "Adaptive real-time trojan detection framework through machine learning," in *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, IEEE, 2016, pp. 120–123.

[113] ——, "Svm-based real-time hardware trojan detection for many-core platform," in *Quality Electronic Design (ISQED), 2016 17th International Symposium on*, IEEE, 2016, pp. 362–367.

[114] A. A. Nasr and M. Z. Abdulmageed, "Automatic feature selection of hardware layout: A step toward robust hardware trojan detection," *Journal of Electronic Testing*, vol. 32, no. 3, pp. 357–367, 2016.

[115] D. M. J. Tax, "One-class classification: Concept learning in the absence of counter-examples.," 2002.

[116] N. H. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective*. Pearson Education India, 2015.

[117] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolić, *Digital integrated circuits: a design perspective*. Pearson Education Upper Saddle River, NJ, 2003, vol. 7.

[118] A. B. Sachid, P. Paliwal, S Joshi, M Shojaei, D Sharma, and V Rao, "Circuit optimization at 22nm technology node," in *2012 25th International Conference on VLSI Design*, IEEE, 2012, pp. 322–327.

[119] C.-L. Cheng, L. N. Nguyen, M. Prvulovic, and A. Zajić, "Exploiting switching of transistors in digital electronics for RFID tag design," *IEEE Journal of Radio Frequency Identification*, vol. 3, no. 2, pp. 67–76, 2019.

VITA

Luong N. Nguyen was born and grew up in Vietnam. He received the B.Sc. degree in Electrical and Computer Engineering from the Hanoi University of Science and Technology in 2013 and the M.Sc. degree in Electrical and Computer Engineering from the Seoul National University in 2016. Since 2016, he has been a Graduate Research Assistant, pursuing the Ph.D. degree in the School of Electrical and Computer Engineering, Georgia Institute of Technology focusing on digital circuit design, software and hardware security, and embedded system. His current research interests span areas of ASIC design, computer architecture, and electrical engineering. He is a past recipient of the 2019 TechConnect Innovation Award, the 2019 second best hardware demo award from the 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), the Korean Government Scholarship Program, and the best paper award from the 2016 Korean SoC Conference.